Secure Moving Object Detection Transformer in Compressed Video with Feature Fusion

Yuru Song, Yike Chen, Peijia Zheng*, Yusong Du and Weiqi Luo
Guangdong Provincial Key Laboratory of Information Security Technology,
MoE Key Laboratory of Information Technology,
School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
*Corresponding author, email: zhpj@mail.sysu.edu.cn

***Abstract*—Moving Object Detection (MOD) is essential for surveillance videos, but the large data volume necessitates significant computational resources. Cloud computing offers a solution, though it raises privacy concerns. In this paper, we propose a secure MOD framework based on the Detection Transformer. The user-side protects video data using format-compatible selective video encryption before storing it in the cloud. In the cloud, we design a method to extract encrypted domain motion information from encrypted videos. Due to the dual constraints of compression and encryption, the extracted information is highly sparse, and common deep learning methods do not perform well. To address this issue, we propose a Convolutional Neural Network (CNN)-Transformer feature enhancement-fusion method to achieve effective feature alignment and fully exploit deep-level motion information. Specifically, we modulate CNN features at multiple scales using Transformer features, where high-level semantic information and low-level spatial information are fused for accurate moving object localization. We evaluate our model on the VIRAT and DUKE-MTMC datasets, demonstrating better detection performance and greater robustness in challenging scenarios compared to previous secure MOD methods.**

I. Introduction

In recent years, visual tasks such as moving object detection (MOD) in videos have become a major focus of artificial intelligence research. Given the extensive and voluminous nature of surveillance video datasets, the necessity for automated MOD is imperative. Local processing of these large datasets can be sluggish, prompting many users to opt for cloud-based solutions. However, uploading unencrypted videos to the cloud poses potential privacy risks. Therefore, it is necessary to implement privacy-enhancing methods on video data before storing it in the cloud. Fig. 1 shows the process of cloud encrypted video processing.

Initial MOD methods [1]–[4] for encrypted domain videos relied on traditional machine learning techniques, resulting in low detection accuracy. With the advancement of deep learning, neural network-based MOD models have greatly outperformed traditional methods. However, due to the disturbance of pixels in encrypted frames, deep learning MOD cannot be directly applied to encrypted videos. Tian et al. [5] proposed the first deep learning-based encrypted domain compressed video MOD framework. Nevertheless, Tian's method has areas for improvement. Firstly, their use of Convolutional Neural Network (CNN) models for MOD, which depends on anchor boxes and complex post-processing steps. Secondly, CNNs primarily capture local object localization features without considering global features, which is inadequate for detecting complex scenes with long-range dependencies. Finally, the motion features extracted from encrypted domain compressed video bitstreams are sparser, and Tian's method did not fully utilize these sparse motion features.

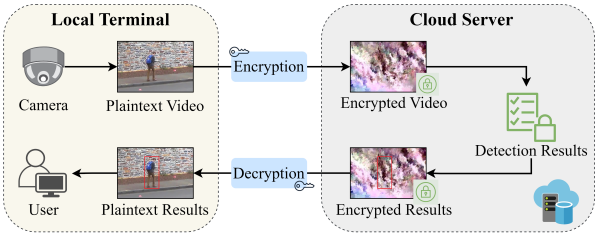


Fig. 1. The cloud-based surveillance system model.

Visual Transformers [6]–[8] have developed rapidly and significantly improved detection performance. However, no research has yet applied Transformers to MOD in encrypted compressed videos. Additionally, CNNs and Transformers learn different types of features—CNNs mainly capture localization features such as edges and lines around objects, while Transformers capture global pixel relationships and high-level semantic features. Many studies have shown that fusing these features benefits various visual recognition tasks [9], [10]. Given the sparsity of motion features extracted from encrypted compressed videos, feature fusion can fully leverage these sparse features, making it important to explore how to enhance model detection performance through this fusion.

In this paper, we design a novel Transformer-based secure MOD framework for compressed videos. Our approach employs the Selective Encryption (SE) technique to encrypt the compressed video bitstream, achieving an optimal balance between encryption efficiency and privacy. We extract encrypted domain motion information by leveraging entropy-encoded syntax elements from the compressed video bitstream and input it into the Deformable-DETR [11] model, using Swin-Transformer [12] as the backbone to extract multi-level features. These features are then fed into the encoder and our CNN-Transformer feature fusion module, with the encoder's

output also serving as input to the feature fusion module, and the final output being fed into the decoder.

The core design of our network is the CNN-Transformer feature enhancement and fusion module, which uses the object localization features from CNN to assist the high-level semantic features from Transformer, thereby increasing feature diversity. Our module has two sub-modules. The spatial enhancement module focuses on the position information of objects, using deformable convolutions to improve the ability to locate moving objects in occluded scenes and scenes with many small objects. The channel fusion module optimizes the channel distribution of features through global pooling and hyperparameter adjustment, dividing the features into multiple groups with semantic information characteristics captured by the encoder, and finally merging the features with the corresponding group's semantic information, achieving effective information fusion within each group. Thus, our module adeptly captures semantic and localization features, enabling comprehensive and efficient inter-domain feature alignment, fully exploiting the sparse information.

Our main contributions can be summarized as follows. **First**, we design a Transformer-based framework specifically for MOD in encrypted compressed videos, utilizing feature enhancement and fusion to fully exploit motion features and address the inherent sparsity of motion information in the encrypted domain. **Second**, we develop a CNN-Transformer feature enhancement-fusion module that captures and fuses high-level semantic and low-level localization features for effective inter-domain feature alignment. **Third**, experimental results on the H.264 and H.265 formatted datasets reveal that our model substantially improves accuracy in challenging scenarios with crowded moving objects.

II. Related Works

Deep-Learning-Based MOD [13], [14] employed CNNs to learn temporal and spatial features from video frames. TransVOD [15] introduced a temporal Transformer to aggregate spatial object queries and feature memories for each frame. However, these models are designed for video frames and are not suitable for the compressed domain. In encrypted videos, pixel values are scrambled and no longer represent accurate information. Existing deep learning MOD models cannot learn effective motion representations from scrambled pixels, making them unsuitable for encrypted videos.

Feature fusion is commonly used in various visual tasks. FPN [16] merges features at different scales to construct multi-layer feature maps, accommodating targets of different sizes. AFF [9] proposed a multi-scale channel attention module to better fuse features from different layers and scales. DA-DETR [17] blended CNN features using Transformer features at multiple scales for accurate object recognition and localization. Unlike these methods, we explore a feature enhancement-fusion method specifically for sparse encrypted compressed information, focusing on the effective utilization of the information. Moreover, we do not limit ourselves to the types of features extracted by the backbone.

Bitstream-encryption-based methods [1]–[4] typically employ format-compatible SE [18] to encrypt videos and then extract syntax elements such as Motion Vector Differences (MVD) from the encrypted compressed video bitstream. These methods, while fast since they do not require decoding compressed videos into frames, rely on traditional machine learning methods with empirical parameter settings for detection, which perform poorly in complex scenarios. To our knowledge, only one piece of research proposes a deep learning-based MOD framework for encrypted compressed video [5]. However, this method uses only a CNN architecture to operate on motion features, insufficiently utilizing sparse information, leaving room for improvement in both detection accuracy and speed.

III. Proposed Secure MOD Scheme with CNN-Transformer Feature Fusion Module

We describe the main procedure of our secure MOD scheme. First, compressed video bitstreams are encrypted locally and then uploaded to a cloud server. In the cloud server, the syntax elements are extracted from the encrypted bitstreams and restructured into motion feature maps. These feature maps serve as the input of our MOD Transformer, which consists of a backbone for extracting feature pyramids, an encoder for obtaining high-level semantic features, a CNN-Transformer feature enhancement-fusion module for feature alignment, and a decoder for annotating moving objects with bounding boxes. Finally, the cloud server sends the MOD results to the local terminal. We consider two popular attack models: Known Ciphertext Model and Known Background Model, and the security proof is the same as [5].

A. Privacy Enhancement on Video

Our secure MOD framework employs a format-compliant SE method [19], [20] for privacy enhancement on H.264 and H.265 videos. This video encryption scheme mainly encrypts intra-prediction modes (IPM), MVD, and residual data. Encrypting these syntax elements ensures significant spatial distortion in the reconstructed frames of the compressed encoding and causes error pixels to propagate from I-frames to reference frames, thereby protecting the visual content's privacy. We use H.265 encoding format as an example, but our framework is applicable to H.264 videos as well.

Intra_4×4 and Intra_16×16 IPM are related to intra-prediction data. For Intra_4×4 IPM, we encrypt the three fixed-length code bits representing the selected mode. For Intra_16×16 IPM, we encrypt the last bit of the codeword to maintain the block encoding mode [18]. MVD and ref_idx are both related to inter-prediction data. Since $suffix$ affects the sign and value of the MVD, we encrypt all $suffix$ bits of the MVD. To maintain format compatibility, only the last bit of ref_idx is encrypted under specific conditions.

Discrete Cosine Transform (DCT) coefficients representing residual data are quantized before entropy coding. We encrypt all $suffix$ bits of the Delta QP during the quantization process. In Context Adaptive Variable Length Coding (CAVLC), we encrypt all sign bits of the $sign\_of\_TrailingOnes$ and

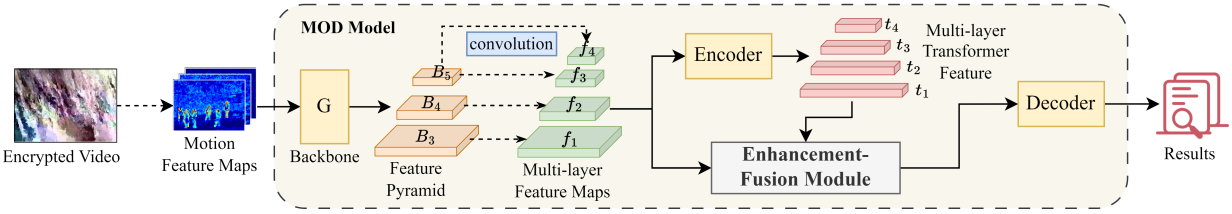


Fig. 2. Overview of the deep learning MOD part of the proposed framework.

$level\_suffix$ codewords. This encryption process secures both the signs and values of the residual data.

B. Syntax Elements Extraction

We denote the t-th frame as $F(t)$, which is divided into a series of 4×4 minimum coding unit blocks. The i-th block is represented as $b_i(t)$. For each $b_i(t)$, the syntax elements contained are represented as $c_i(t)$, $m_i(t)$ and $d_i(t)$, corresponding respectively to the consumed bits, MVD magnitude, and residual density data. These syntax elements are not affected by SE and can reflect certain motion information.

The consumption of coding bits can reflect the degree of pixel changes caused by unpredictable motion, thereby reflecting motion information. Assuming that a basic coding unit block $B(t)$ contains θ 4×4 blocks, $c_i(t) = \lfloor \frac{\texttt{cit}(B(t))\times 4}{\sqrt{\theta}} \rfloor, i \in \mathbb{I}(t)$, where $\texttt{cit}(\cdot)$ is a function calculating the bit count of the coding unit and $\mathbb{I}(t) = \{i_1, i_2, \cdots, i_c\}$ represents the set of addresses of all 4×4 blocks within $B(t)$.

Motion Vector (MV) can reflect motion information and encoded as predictive MVs and MVD, but only MVD is entropy coded in video compression standards. Therefore, we use MVD to estimate motion information. $m_i(t)$ is defined as $m_i(t) = \lfloor \sqrt{L_{\mathrm{x}}(t)^2 + L_{\mathrm{y}}(t)^2} \rfloor$, where $L_{\mathrm{x}}(t)$ and $L_{\mathrm{y}}(t)$ are the lengths of the $suffix$ codeword for the horizontal and vertical components of the MVD, respectively. However, I-frames are encoded independently and lack MVs, which can cause confusion and temporal inconsistency in the model. To address this, we interpolate MVs into I-frames using Inverse Distance Weighted (IDW) temporal interpolation, leveraging the strong spatial-temporal correlation of adjacent frames.

Given that the contour areas of moving objects undergo a high degree of pixel substitution, these areas exhibit higher density compared to background blocks, enabling the density of non-zero DCT residuals $d_i(t)$ to highlight the edges of moving objects. We use $\texttt{dit}(\cdot)$ to denote the function that calculates the number of non-zero coefficients in a DCT block. Supposing a DCT block comprises δ 4×4 blocks, the residual density $d_i(t)$ is defined as $d_i(t) = \lfloor \frac{\texttt{dit}(B_{dct}(t))}{\sqrt{\delta}} \rfloor$.

Subsequent to the extraction of syntax elements, we reshape $c_i(t)$, $m_i(t)$, and $d_i(t)$ into two-dimensional matrices and collectively refer to them as motion feature maps.

C. Deep Learning MOD Model using Detection Transformer

Since Deformable-DETR contains a feature pyramid structure, it can better focus on targets of different scales and achieve feature fusion to a certain extent. In addition, Deformable-DETR uses the deformable attention mechanism to perform calculations in local areas of interest, which can more flexibly adapt to changes in target shape and improve the perception ability of the model in different scenarios. Therefore, we use Deformable-DETR as our basic model to extract the feature pyramid.

Our model consists of a backbone for feature extraction, an encoder containing 6 encoder-layers, a CNN-Transformer feature enhancement-fusion module, and a decoder with 6 decoder-layers for final predictions. Each encoder-layer and decoder-layer contains a multi-head deformable attention mechanism, residual connections and layer normalization, and a feed forward neural network. As shown in the Fig. 2, the feature pyramid B_3-B_5 extracted from the Swin-Transformer backbone is transformed into multi-scale feature maps f_i $(i = 1, 2, 3, 4)$ through a series of convolutions. Specifically, B_3-B_5 are transformed into f_1-f_3 via 1×1 convolutions, while the lowest resolution feature map f_4 is obtained from B_5 using a stride-2 3×3 convolution. Then, f_1-f_4 is used as the input of the encoder to obtain the Transformer features t_1-t_4. Next, the CNN-Transformer enhancement-fusion module takes the CNN features f_i $(i = 1, 2, 3, 4)$ and the Transformer features t_i $(i = 1, 2, 3, 4)$ as input. An overview of the processing on the encrypted video is shown in Fig. 2.

D. CNN-Transformer Feature Enhancement-Fusion Module

Syntax elements, extracted from encrypted domain compressed video bitstreams, contains sparser information compared to original RGB frames. Fully utilizing this motion information is challenging. Previous studies[9], [17] have shown that feature fusion can improve feature utilization and detection accuracy. Therefore, we add a CNN-Transformer feature enhancement-fusion module to Deformable-DETR. Notably, our module is plug-and-play and can be integrated into any DETR-like model. As illustrated in Fig. 3, the module takes multi-scale motion feature maps and multi-scale Transformer features as input. It comprises two parts: a spatial enhancement module for object localization and a channel fusion module for intra-group alignment.

1) Spatial-Enhancement Module: This module performs convolution operations on the multi-scale feature maps f_i $(i = 1, 2, 3, 4)$ to focus on regions related to moving objects, enhancing the ability to locate objects in occluded scenes and those with many small objects. We use deformable convolution v2 [21] for sparse learning of low-level localization features, allowing adaptive adjustment of convolution kernel sampling

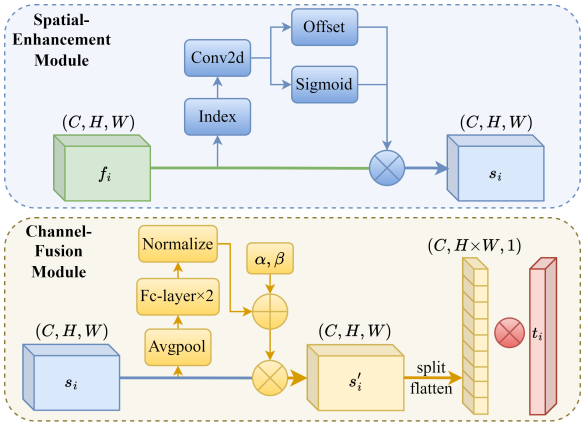


Fig. 3. Overview of proposed CNN-Transformer feature enhancement-fusion module.

positions to capture object location features more accurately. This enhances the ability to locate moving objects in occluded and densely populated scenes. We focus on the motion of each object, learning offsets based on their spatial positions, and then fusing the features of the same moving object:

$$s_i = \sum_{n=1}^{N} \omega_n \cdot f_i(p_n + O_{p_n}, c) \cdot \triangle m_n \tag{1}$$

where s_i is the new CNN feature obtained from this module, with each layer's feature dimension matching the original multi-scale feature map. N is the number of coefficient sampling positions, $p_n + O_{p_n}$ is the offset position of the target at p_n, and $\triangle m_n$ is the self-learned importance scalar of p_n.

2) Channel-Fusion Module: This module first adjusts the features of $s_i \in \mathbb{R}^{C \times H_i \times W_i}$ $(i = 1, 2, 3, 4)$ based on task importance within the channel, then groups CNN features according to the number of Transformer feature groups obtained from the encoder, and aligns and fuses each corresponding CNN-Transformer feature group.

To clarify, let's consider one layer of CNN feature s_1 and its corresponding Transformer feature t_1. We dynamically adjust the weights of s_1's channels in the context of the current detection task. Specifically, we first perform global average pooling to reduce complexity, then use two fully connected layers with ReLU activations and a normalization layer. Finally, we use a hard-sigmoid function to normalize the output weights to the range [-1,1] and adjust the channel features using four hyperparameters:

$$s'_1 = \max\left[\alpha^1(s_1) \cdot s_{1_c} + \beta^1(s_1), \alpha^2(s_1) \cdot s_{1_c} + \beta^2(s_1)\right] \tag{2}$$

where $\left[\alpha^1, \beta^2, \alpha^2, \beta^2\right]$ are hyperparameters controlling the adjustment of channel weights.

Next, we flatten the processed CNN features. Noting that the dimension of t_1 is $C \times (H_1 \times W_1) \times 1$, divided into k equal-sized image blocks along the $H_1 \times W_1$ dimension. We flatten s'_1 and perform feature fusion on the corresponding image blocks:

$$V_1 = \texttt{flatten}(s'_1) + t_1 \tag{3}$$

TABLE I
ABLATION EXPERIMENTS ON FEATURE ENHANCEMENT-FUSION MODULE.

SEM	CFM	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
×	×	64.5	95.9	77.9	6.6	50.5	68.1
✓	×	66.7	95.9	80.9	6.6	52.2	70.3
×	✓	66.7	95.9	80.9	6.6	52.9	70.3
✓	✓	**67.6**	**96.2**	**82.6**	**6.7**	**53.7**	**71.2**

Finally, the aligned features V_i $(i = 1, 2, 3, 4)$ are used as input for the decoder.

IV. EXPERIMENT

A. Experiment Setups

Consistent with previous works[5], we established datasets for our experiment from two large HD surveillance video datasets: VIRAT and Duke-MTMC. VIRAT consists of 680 video clips containing 400,806 frames, while Duke-MTMC includes 1,109 clips with 315,625 frames. We randomly selected 70% of the video clips for training, 5% for validation, and 25% for testing. We used the standard COCO Average Precision (AP) accuracy metrics [22] to evaluate performance. For loss calculation, we used L1 Loss with a weight of 5 for bounding box loss, GIoU Loss with a weight of 2 for IoU loss, and Focal Loss with a weight of 2 for bounding box classification. The number of object queries was set to 300. We used four NVIDIA GeForce RTX 3090 GPUs for model training and testing. The training of our two-stage model was optimized using the AdamW optimizer, with an initial learning rate of 0.0002 and a weight decay of 0.0001.

B. Ablation Study

The proposed CNN-Transformer feature enhancement-fusion module consists of the spatial-enhancement module (SEM) and the channel-fusion module (CFM). We conducted experiments on the Duke-MTMC dataset to examine their contributions to our MOD model's performance. Since SEM primarily enhances Transformer features using CNN methods, we used the fusion methods in CFM for feature integration during the experiments. As demonstrated in Table I, adding either module improved the model's detection results. The improvements from both modules were remarkably similar, but adding CFM provided a greater boost in AP_M, likely because channel fusion enhances the detection of medium-sized objects. The combination of SEM and CFM achieved the best 67.6% mAP, indicating that these structures are complementary.

C. Comparison Study

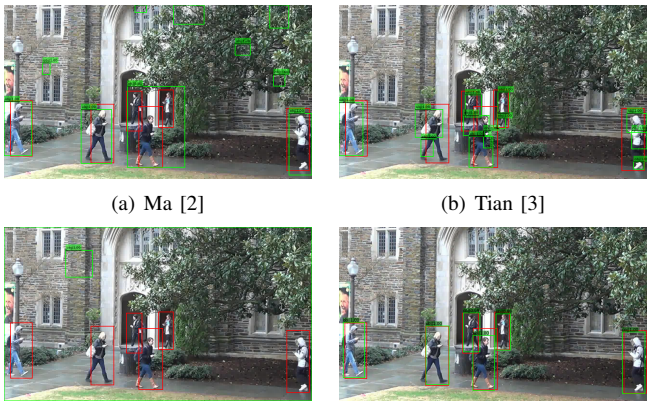
1) Comparison with Existing Secure MOD Methods: We compared our approach with secure MOD methods proposed by Guo [1], Ma [2], Tian [3], Liu [4], and Tian [5]. The first four methods use traditional machine learning for object detection, while the last one employs deep learning. To ensure fairness, we used scenario-independent testing, excluding *VIRAT_S_0400* from our training set. The results are shown in Table II. Due to the absence of large objects in the

TABLE II
THE COMPARISON WITH ALL EXISTING TRADITIONAL METHODS ON THE SCENE OF *VIRAT_S_0400*.

Method	mAP	AP_{50}	AP_{75}	AP_S	AP_M
Guo [1]	11.0	44.1	0.90	5.80	18.3
Ma [2]	13.6	60.9	1.20	15.5	11.0
Tian [3]	16.4	54.9	5.70	15.3	18.0
Liu [4]	16.1	70.6	1.60	17.6	14.1
Tian [5]	50.9	96.9	46.7	**46.6**	54.5
Ours	**51.2**	**97.0**	**47.3**	46.2	**55.3**

TABLE III
THE COMPARISON OF OUR METHOD AND TIAN [5] ON THE FULL DATASET.

Dataset	Method	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Duke-MTMC	Tian [5]	66.1	95.0	79.9	6.5	51.3	69.7
	Ours	**67.6**	**96.2**	**82.6**	**6.7**	**53.7**	**71.2**
VIRAT	Tian [5]	36.2	78.5	28.1	22.1	41.3	**59.2**
	Ours	**36.4**	**79.1**	**28.4**	**22.4**	**41.8**	58.6



(a) Ma [2] (b) Tian [3]

(c) Liu [4] (d) Ours

Fig. 4. An example of visualization of results on the *camera_7_0022* from Duke-MTMC dataset. The red boxes represent the ground-truth and the green boxes indicate the results. The results from traditional methods are derived from their detection algorithms, and we use a confidence score threshold of 0.5 to display these bounding boxes.

scene, the result of AP_L is -1.000. Additionally, we compared our method with existing deep learning MOD methods on the complete VIRAT and Duke-MTMC datasets, as shown in Table III. Our approach achieved the highest detection performance on the *VIRAT_S_0400* video clip, except for AP_S, and showed significant improvements over traditional methods. On the complete Duke-MTMC and VIRAT datasets, our method outperformed Tian's [5] method, which used a CNN-based approach. Our DETR-based deep learning method, which integrates CNN and Transformer features, demonstrated superior overall performance.

To further showcase our method's superiority, we conducted comparative experiments in challenging scenarios, such as crowded and dynamic background scenes, and visualized the detection results on plaintext frames. As exemplified in Fig. 4, our method's detection performance in overlapping target scenarios is far superior to other methods.

2) Comparison with Existing MOD Models: Based on Tian's [5] comparative experiments, we compared our approach with MotionRec and T_RetinaNet proposed in [13] on the datasets used in the respective paper. We encrypted the dataset videos and split them into training and testing sets according to the paper's standards. We changed the detection categories to three (background, person, vehicle) and used the same evaluation metrics. We conduct experiments in all scenarios and present the results for three of them. As shown in Table IV, when MotionRec's depth was 30, the plaintext domain model achieved the best detection results. Although our method did not achieve the highest AP value compared to existing plaintext MOD methods, this is because plaintext MOD methods can directly access more motion information. In contrast, our method accesses limited motion information from encrypted video bitstreams due to privacy protection requirements.

However, it is noteworthy that our performance still surpassed three other plaintext MOD methods and was better than Tian's [5] deep learning MOD method. The slight difference between our method and the best plaintext method validates our framework's effectiveness and superiority in encrypted video MOD tasks. Additionally, MotionRec's prediction speed is only 2-5 fps, while our model achieves a detection speed of 42.2 fps, as it operates on compressed domain videos without needing to decode them into frames.

TABLE IV
THE AP_{50} COMPARISON BETWEEN MOD MODELS. T_R REPRESENTS T_RETINANET, MR STANDS FOR MOTIONREC. D. DENOTES THE DATA DOMAIN WHICH INCLUDES P.(PLAINTEXT) AND E.(ENCRYPTED) DOMAIN.

Method	D.	*Sofa*	*Parking*	*Bungalows*	Overall
T_R_V1	P.	$72.0_{-2.2}$	$47.2_{-21.8}$	$88.7_{+2.5}$	$69.3_{-7.6}$
T_R_V2	P.	$35.1_{-39.1}$	$30.0_{-39.0}$	$42.6_{-43.8}$	$26.0_{-50.9}$
MR_V1	P.	$\mathbf{80.5}_{+6.3}$	$\mathbf{69.5}_{+0.5}$	$\mathbf{89.0}_{+2.8}$	$\mathbf{79.7}_{+2.8}$
MR_V2	P.	$70.3_{-3.9}$	$61.2_{-7.8}$	$84.6_{-1.6}$	$72.0_{-4.9}$
Tian[5]	E.	$74.2_{-0.0}$	$68.8_{-0.2}$	$86.0_{-0.2}$	$76.3_{-0.6}$
Ours	E.	74.2	69.0	86.2	76.9

D. Adaptation to Other Video Codec and Encryption

Our method is not dependent on the video's encoding format or the selective encryption method used. We conducted experiments with H.264 and H.265 videos encrypted using different methods, as shown in the Table. The results indicate that the detection outcomes remain consistent regardless of the encryption method used for the same video encoding format. This consistency is because the encryption scheme does not affect the syntax elements in the same encoded video stream, resulting in identical extracted features. Additionally, detection results slightly differ between H.264 and H.265 videos due to compression efficiency differences, with H.265 yielding sparser information, leading to slightly lower detection performance compared to H.264.

V. CONCLUSIONS

We propose a Transformer-based deep learning secure MOD framework for encrypted domain compressed videos. Our approach uses SE to encrypt video bitstreams and then extracts motion information directly from the encrypted video,

TABLE V
PERFORMANCE COMPARISON OF THE PROPOSED FRAMEWORK USING VARIOUS ENCRYPTION SCHEMES APPLIED TO H.264/H.265 VIDEOS.

Encryption Scheme	Video Codec	VIRAT			Duke-MTMC		
		mAP	AP_{50}	AP_{75}	mAP	AP_{50}	AP_{75}
Wang [23]	H.264	37.1	79.2	30.3	67.9	96.3	82.8
Xu [18]	H.264	37.1	79.2	30.3	67.9	96.3	82.8
Shahid [19]	H.265	36.4	79.1	28.4	67.6	96.2	82.6
Sallam [24]	H.265	36.4	79.1	28.4	67.6	96.2	82.6

which is used as input to the DETR model. We employ a CNN-Transformer feature enhancement-fusion method for effective feature alignment, improving the utilization of sparse information. Experimental results demonstrate that our method achieves advanced performance compared to other secure MOD techniques, with superior detection capabilities.

ACKNOWLEDGMENT

This work is supported in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515030087 and Grant 2022A1515011512, the National Natural Science Foundation of China under Grant 62272498, and the Guangdong Provincial Key Laboratory of Information Security Technology (No. 2023B1212060026). Thanks are due to Dr. Xianhao Tian for assistance with the experiments.

REFERENCES

[1] J. Guo, P. Zheng, and J. Huang, "An efficient motion detection and tracking scheme for encrypted surveillance videos," *ACM Trans. Multimedia Comput. Commun. & Appl.*, Sep. 2017.

[2] X. Ma, H. Peng, H. Jin, and B. Zhu, "Privacy-preserving cloud-based video surveillance with adjustable granularity of privacy protection," in *Proc. ICIP*, 2018.

[3] X. Tian, P. Zheng, and J. Huang, "Robust privacy-preserving motion detection and object tracking in encrypted streaming video," *IEEE Trans. Inf. Forensics Security*, pp. 1–1, 2021.

[4] C. Liu, X. Ma, S. Cao, J. Fu, and B. B. Zhu, "Privacy-preserving motion detection for HEVC-compressed surveillance video," *ACM Trans. Multimedia Comput. Commun. & Appl.*, p. 27, 2022.

[5] X. Tian, P. Zheng, and J. Huang, "Secure deep learning framework for moving object detection in compressed video," *IEEE Trans. Depend. Sec. Comput.*, 2023.

[6] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. ECCV*, 2020, pp. 213–229.

[7] D. Han, X. Pan, Y. Han, S. Song, and G. Huang, "Flatten transformer: Vision transformer using focused linear attention," in *Proc. ICCV*, 2023, pp. 5961–5971.

[8] C. Zhao, Y. Sun, W. Wang, *et al.*, "Ms-detr: Efficient detr training with mixed supervision," in *Proc. CVPR*, 2024, pp. 17 027–17 036.

[9] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard, "Attentional feature fusion," in *Proc. WACV*, 2021.

[10] Y. Cheng, R. Cai, Z. Li, X. Zhao, and K. Huang, "Locality-sensitive deconvolution networks with gated fusion for rgb-d indoor semantic segmentation," in *Proc. CVPR*, 2017, pp. 3029–3037.

[11] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[12] Z. Liu, Y. Lin, Y. Cao, *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. ICCV*, 2021, pp. 10 012–10 022.

[13] M. Mandal, L. K. Kumar, M. S. Saran, and S. K. vipparthi, "MotionRec: A unified deep framework for moving object recognition," in *Proc. WACV*, Mar. 2020, pp. 2723–2732.

[14] A. Guzman-Pando and M. I. Chacon-Murguia, "Deep-FoveaNet: Deep fovea eagle-eye bioinspired model to detect moving objects," *IEEE Trans. Image Process.*, pp. 7090–7100, 2021.

[15] L. He, Q. Zhou, X. Li, *et al.*, "End-to-end video object detection with spatial-temporal transformers," in *Proc. ACM MM*, 2021, pp. 1507–1516.

[16] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. CVPR*, 2017, pp. 2117–2125.

[17] J. Zhang, J. Huang, Z. Luo, G. Zhang, X. Zhang, and S. Lu, "Da-detr: Domain adaptive detection transformer with information fusion," in *Proc. CVPR*, 2023, pp. 23 787–23 798.

[18] D. Xu, R. Wang, and Y. Q. Shi, "Data hiding in encrypted H. 264/AVC video streams by codeword substitution," *IEEE Trans. Inf. Forensics Security*, pp. 596–606, 2014.

[19] Z. Shahid and W. Puech, "Visual protection of HEVC video by selective encryption of CABAC binstring," *IEEE Trans. Multimedia*, pp. 24–36, 2014.

[20] Q. Sheng, C. Fu, Z. Lin, *et al.*, "A fast selective encryption scheme for h. 264/avc video with syntax-preserving and zero bit rate expansion," *Signal Image & Video Process.*, pp. 1–15, 2023.

[21] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *Proc. CVPR*, 2019, pp. 9300–9308.

[22] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, Springer, 2014, pp. 740–755.

[23] Y. Wang, M. O'Neill, and F. Kurugollu, "A tunable encryption scheme and analysis of fast selective encryption for CAVLC and CABAC in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1476–1490, 2013.

[24] A. I. Sallam, O. S. Faragallah, and E. M. El-Rabaie, "HEVC selective encryption using RC6 block cipher technique," *IEEE Trans. Multimedia*, pp. 1636–1644, 2018.