Context Enhanced Transformer for Single Image Object Detection in Video Data

Seungjun An*[1], Seonghoon Park*[1], Gyeongnyeon Kim*[1], Jeongyeol Baek[2], Byeongwon Lee[2], Seungryong Kim[1]

[1]Korea University, Seoul, Korea
[2]SK Telecom, Seoul, Korea
{dkstmdwns, seong0905, kkn9975}@korea.ac.kr, {jeongyeol.baek, bwon.lee}@sk.com, seungryong_kim@korea.ac.kr

Abstract

With the increasing importance of video data in real-world applications, there is a rising need for efficient object detection methods that utilize temporal information. While existing video object detection (VOD) techniques employ various strategies to address this challenge, they typically depend on locally adjacent frames or randomly sampled images within a clip. Although recent Transformer-based VOD methods have shown promising results, their reliance on multiple inputs and additional network complexity to incorporate temporal information limits their practical applicability. In this paper, we propose a novel approach to single image object detection, called Context Enhanced TRansformer (CETR), by incorporating temporal context into DETR using a newly designed memory module. To efficiently store temporal information, we construct a class-wise memory that collects contextual information across data. Additionally, we present a classification-based sampling technique to selectively utilize the relevant memory for the current image. In the testing, We introduce a test-time memory adaptation method that updates individual memory functions by considering the test distribution. Experiments with CityCam and ImageNet VID datasets exhibit the efficiency of the framework on various video systems. The project page and code will be made available at: https://ku-cvlab.github.io/CETR.

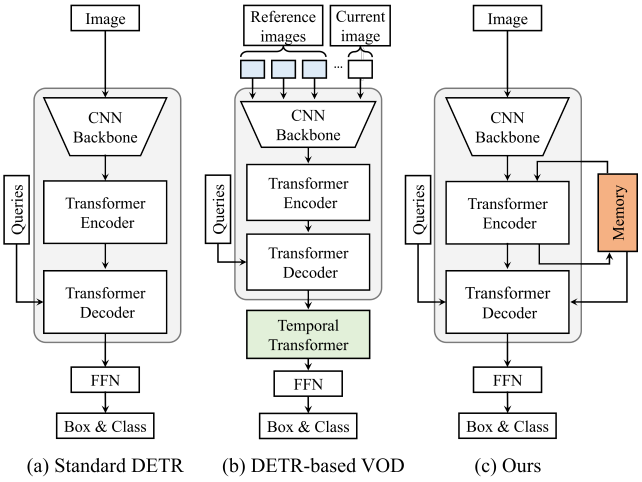

Figure 1: Comparisons between existing works and ours. (a) standard DETR (Carion et al. 2020), (b) DETR-based video object detection (Zhou et al. 2022), and (c) our proposed framework, dubbed CETR. Our method effectively detects objects in video data without adding heavy components.

Introduction

Object detection is one of the fundamental and essential tasks in the computer vision field with its extensive versatility across a wide range of applications. Moreover, various applications in real-world scenarios, including video surveillance (Nascimento and Marques 2006; Fu et al. 2019), autonomous driving (Chen et al. 2015, 2016), and robot navigation (Hernández et al. 2016), heavily rely on video data. Despite the remarkable success of object detectors for a single image, directly applying them to video data encounters challenges due to appearance deterioration caused by motions and occlusions.

To address this challenge, video object detection (VOD) models (Zhu et al. 2017a; Wang et al. 2018) have been proposed to improve object detection performance by leveraging temporal information. Previous approaches usually aggregate features from nearby frames exploiting optical flow (Zhu et al. 2017a; Wang et al. 2018) or LSTM (Kang et al. 2017a,b). Nevertheless, these methods primarily focus on short-term frames, thus limiting their ability to capture a more extensive feature representation. To overcome this limitation, attention-based approaches (Chen et al. 2020; Deng et al. 2019a,b) attempt to capture long-range temporal dependency by utilizing memory structures to aggregate features globally or locally. Yet, they depend on randomly sampled images within a clip, which struggle to integrate holistic contextual information from video data. Furthermore, the construction of stacked memory modules to store features of adjacent frames incurs high computational costs and unnecessary memory usage.

On the other hand, in light of the remarkable performance of Transformer-based models in image object detection (Carion et al. 2020; Liu et al. 2022), e.g., detection with Transformers (DETR), researchers have commenced extending them to the video domain (Zhou et al. 2022; Wang et al. 2022a). However, these methods exhibit an essential

*These authors contributed equally.

reliance on auxiliary networks and the need for multiple sequential frames, as shown in Fig. 1. Such prerequisites lead to a considerable decrease in processing speed, thereby failing to fulfill the real-time operational demands of various systems. As a consequence, there is a need for more efficient and streamlined approaches that can meet the real-time requirements essential for practical applications.

In this work, we propose a novel single image object detection method, dubbed Context Enhanced TRansformer (CETR), that effectively incorporates contextual information across the given data. Following the recent trend of Transformer-based detectors, we adopt DETR as our baseline. Due to the inherent attention mechanism within the Transformer framework, our approach effectively incorporates temporal information through attention modules. In order to utilize temporal context without requiring additional reference frames or networks, we present a context memory module (CMM) that stores class-wise feature representations and updates effectively using momentum update in a non-parametric manner. The memory also represents each class as a set of prototypes, allowing intra-classes to contain a variety of attributes. In addition, to effectively capture relevant information for the current features, we introduce a score-based sampling methodology. By propagating the encoded memory features through a classification network for making predictions, CETR employs a sampled class-specific memory that closely aligns with the current input. Furthermore, we introduce an adaptive memory updating technique tailored to the test domain across different camera settings. Unlike the uniform exponential moving average update employed during training, we implement an online updating strategy aligned with the class-wise distribution of the test domain. Utilizing a weighted sum of the target and source domain memories, this strategy facilitates adaptation toward the test data distribution while retaining contextual information from the training phase.

To validate the effectiveness of the proposed method, we conduct extensive experiments on the CityCam dataset (Zhang et al. 2017), one of the real traffic video data. Furthermore, experiments on the ImageNet VID (Russakovsky et al. 2015) demonstrate that our framework achieves comparable accuracy with the state-of-the-art video object detectors with a much faster speed and efficient memory resource. We also perform detailed ablation studies and deeply analyze the memory module to confirm that it is effective at capturing contextual information.

Related Work

Single image object detection Single image object detectors have been extensively explored due to the development of deep convolutional neural networks (CNNs). CNN-based object detectors can be classified into two pipelines: two-stage and one-stage detectors. Two-stage detectors (Girshick 2015; Ren et al. 2015; Dai et al. 2016) generate coarse object proposals and then classify the proposals and regress the bounding boxes to refine them. In contrast, one-stage detectors (Duan et al. 2019; Tian et al. 2019) directly predict object locations and categories in an image by utilizing densely designed anchors. In recent years, DETR (Carion et al. 2020), a prominent Transformer-based object detector, casts object detection as a direct set prediction problem by removing hand-crafted representations and post-processing techniques. Many follow-up works (Li et al. 2022; Liu et al. 2022; Meng et al. 2021; Zhu et al. 2020) have attempted to address the slow training convergence of DETR's inefficient design and use of queries. In this paper, we choose these variants of DETR as our baseline considering this efficiency.

Video object detection. Video object detection (VOD) methods aim to address the challenging cases, such as motion blur, and occlusion, suffered from the single frame. To tackle this problem, many studies have focused on improving the performance of the current frame by leveraging temporal information across videos. For example, FGFA (Zhu et al. 2017a), MANET (Wang et al. 2018), and THP (Zhu et al. 2018) utilize optical flow derived from FlowNet (Dosovitskiy et al. 2015) by aligning and aggregating the nearby features from current frames. TPN (Kang et al. 2017a) and TCNN (Kang et al. 2017b) exploit LSTM (Hochreiter and Schmidhuber 1997) to construct temporal coherence between detected bounding boxes. To capture long-range dependencies, numerous methods adopt self-attention mechanism. Among them, SELSA (Wu et al. 2019) presents to use of global temporal cues by taking the full-sequence level feature aggregation. OGEMN (Deng et al. 2019a) proposes to use object-guided external memory for further global aggregation. MEGA (Chen et al. 2020) presents a memory module that considers aggregating global and local information to enhance the feature representation. Recently, TransVOD (Zhou et al. 2022) extends the DETR detector into the video object detection domain via a temporal Transformer.

Test-time adaptation. Test-time adaptation (TTA) attempts to adapt pre-trained models to test data without relying on source domain data or incurring labeling costs. Existing TTA methods (Wang et al. 2020, 2022b; Li et al. 2016) typically recalibrate batch normalization (BN) layers using a batch of test samples. However, since Transformers typically do not contain a BN layer, it is not appropriate to apply the re-estimating BN statistics method to Transformer-based models. Alternatively, several studies (Chen et al. 2022; Iwasawa and Matsuo 2021; Jang and Chung 2022) adopt pseudo labels generated at test time for updating the model. In the domain of object detection, TTAOD (Chen et al. 2023) focuses on enhancing real-time robustness across target domains via self-training and feature distribution alignment. This paper introduces a test-time adaptation technique suitable for Transformer-based image detectors, leveraging a newly designed memory module.

Methodology

In this section, we first review DETR framework (Carion et al. 2020), and then introduce our proposed framework, called CETR, which is a single frame context-aware object detector with context memory module (CMM), score-based sampling strategy, and memory-guided Transformer decoder (MGD) in detail.

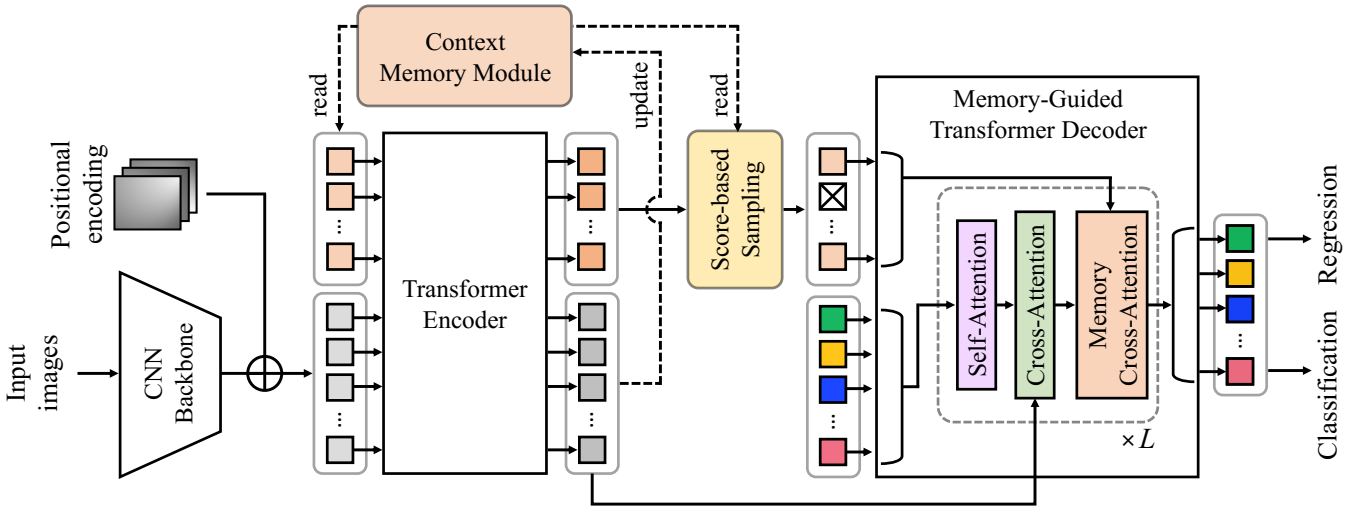

Figure 2: Overview of our framework. CETR builds upon the DETR (Carion et al. 2020) architecture. Within our framework, a pivotal component is the context memory module (CMM), which serves as the input for the Transformer encoder. Subsequently, the encoded memory features are passed through the classification network. Predicted probability serves as a threshold for score-based sampling. The sampled class-wise memory is aggregated with the query using the cross-attention mechanism within the memory-guided Transformer decoder (MGD).

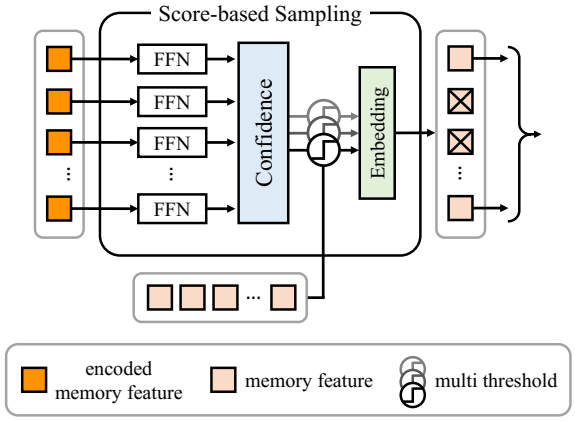

Figure 3: Details of score-based sampling module.

Preliminaries: Revisiting DETR

DETR and its variants are based on encoder-decoder Transformer architecture. The encoder layer consists of a multi-head self-attention and feed-forward network (FFN), and the decoder layer has additional cross-attention layers. Specifically, given an input image I, CNN backbone extract feature map $F \in \mathbb{R}^{H \cdot W \times d}$, where d denotes the dimension of feature and H, W are the height and width of the feature map, respectively. Then F augmented with positional encoding are fed into the Transformer encoder (denoted by $\text{Enc}(\cdot)$):

$$\mathcal{F} = \text{Enc}(F). \tag{1}$$

Note that we omit the positional encoding in the description for clarity. $\text{Enc}(\cdot)$ is composed of self-attention layers, which would be applied to F to generate the query Q, key K, and value V vectors for exchanging information features at all spatial positions. Self-attention of the Transformer encoder is conducted as:

$$\text{Attn}(Q = F,\ K = F,\ V = F), \tag{2}$$

where multi-head attention is represented as:

$$\text{Attn}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d}})V. \tag{3}$$

The image feature $\mathcal{F}$ is input to the Transformer decoder, with object queries $\mathcal{O}$. The Transformer decoder is composed of the following two types of attention layers: multi-head self-attention and multi-head cross-attention.

$$\mathcal{O}^l_{\text{sa}} = \text{Self-Attn}(Q = \mathcal{O}^l,\ K = \mathcal{O}^l,\ V = \mathcal{O}^l), \tag{4}$$

$$\mathcal{O}^{l+1} = \text{Cross-Attn}(Q = \mathcal{O}^l_{\text{sa}},\ K = \mathcal{F},\ V = \mathcal{F}), \tag{5}$$

where decoder blocks are repeated L times and $\mathcal{O}^l$ means object queries of l-th decoder block. Then, the final object queries $\mathcal{O}^{L+1}$, which have acquired semantic information from image features, are passed through a feedforward neural network (FFN) for classification and box regression. Finally, by the Hungarian algorithm, one-to-one matching between predicted objects and their corresponding ground-truth targets is established. We propose a memory module that can be applied to single frame DETR-like methods for processing video data.

Our Approach: CETR

Overview. Most VOD methods that utilize memory modules or reference images have a large memory footprint, which limits the amount of information that can be used at

one time. To alleviate this, recent methods randomly sample memory (Deng et al. 2019a; Chen et al. 2020) or utilize information from frames close to the current frame (Zhou et al. 2022; Cui 2023). However, these approaches also have the drawback of either requiring information from the entire frame beforehand or being able to reference unnecessary information. To address these limitations, we propose CETR that selectively stores only the necessary information from the data to form a class-specific memory with a fixed size, and utilizes only the information useful for the current frame in memory. This method enables models to efficiently leverage contextual information from the entire dataset for each frame. For this, as illustrated in Fig. 2, we introduce three modules applicable to single frame DETR-like methods for the video data: 1) the context memory module (CMM), which stores contextual information of the entire dataset in a fixed size; 2) the score-based sampling, which samples only the necessary information from memory for the current frame data; and 3) the memory-guided Transformer decoder (MGD), which enhances the semantic information of object queries using the sampled spatio-temporal memory information.

Context Memory Module. Most single frame DETR-like methods employ a Transformer encoder to aggregate spatial information from current image features, enabling each image feature to include spatial context information from the current image. However, when the CNN backbone extracts ambiguous features from the current image due to challenges such as low image quality or part occlusion, they may struggle to effectively refine the image features. To mitigate this limitation, We use the fixed size memory obtained from the entire dataset to enhance each single frame image feature using temporal context information, without the need to directly use data from other frames.

Our proposed CMM has a multi-prototype class-wise memory $M \in \mathbb{R}^{C \cdot K \times d}$ with K prototypes for each of the C classes. In our pipeline, the feature map F and M would be fed into the Transformer encoder:

$$[\mathcal{F}, \mathcal{M}] = \text{Enc}([F, M]), \tag{6}$$

where $[\cdot, \cdot]$ means concatenation. In the Transformer encoder, the current information of F and the spatio-temporal contextual information of M are aggregated. Consequently, image feature $\mathcal{F}$ gains rich contextual information from M, and simultaneously, encoded memory feature $\mathcal{M}$ obtains class information fitted to the current image. Then, $\mathcal{M}$ is passed to the score-based sampling module to obtain the classification score of the current image, and $\mathcal{F}$ is forwarded to the Transformer decoder similar to DETR.

$\mathcal{F}$ is also utilized in the context memory module for memory update. The context memory module extracts N instance features $\tilde{\mathcal{F}} = \{f_n\}_{n=1}^{N}$ from image feature $\mathcal{F}$ and the set of class-wise memory $M = \{m_{c,k}\}_{c,k=1}^{C,K}$ is updated as:

$$m_{c,k_n} \leftarrow \alpha m_{c,k_n} + (1-\alpha) f_n, \tag{7}$$

$$\text{where } k_n = \text{argmax}_k \{\langle f_n, m_{c,k} \rangle\}_{k=1}^{K}, \tag{8}$$

where $\langle \cdot, \cdot \rangle$ is defined as correlation between two features, and $\alpha \in [0, 1]$ is a momentum coefficient. During training, $\tilde{\mathcal{F}}$ is extracted from the ground-truth box, while during inference, $\tilde{\mathcal{F}}$ is extracted from the predicted box. Aggregated $\mathcal{F}$ with M in the Transformer encoder contains abundant contextual information. Accordingly, M updated recurrently by $\mathcal{F}$ in each image of the video dataset acquires contextual information about the entire dataset that was previously observed. Additionally, the class-wise memory with multi-prototypes can also accommodate diverse distributions of instance features appearing in the entire dataset.

Figure 4: Visualization of the attention map. For certain classes, we present an attention map showing the correlation between class-wise memory and current image features.

Score-based Sampling. Carefully selecting the relevant information from the memory is equally important as creating a high-quality memory. However, recent VOD works that employ memory modules often fail to guarantee optimal memory sampling by either randomly sampling memory or utilizing only the memory information around the current frame. Empirically, we have discovered that utilizing class information from the current image to selectively sample memory leads to significant performance improvement. Detailed results of the related experiments are provided in Table 5. Motivated by this, we introduce a score-based sampling module to extract information relevant to the current image from the class-wise memory M (see Fig. 3). The score-based sampling module consists of two parts: a classification part and a multi-threshold sampling part. In the classification part, the sampling module obtains classification score $p_{c,k} \in [0, 1]$ by passing encoded memory feature $\mathcal{M} = \{\mathfrak{m}_{c,k}\}_{c,k=1}^{C,K}$, which contains the class information of the current image. This operation is executed via a classification head $\text{FFN}_{c,k}$ composed independently for each $\mathfrak{m}_{c,k}$, followed by a sigmoid function:

$$p_{c,k} = \text{Sigmoid}(\text{FFN}_{c,k}(\mathfrak{m}_{c,k})). \tag{9}$$

Subsequently, in the multi-threshold sampling part, the sampled memory $\tilde{M} = \{\tilde{m}_{c,k}\}_{c,k=1}^{C,K}$ is obtained by:

$$\tilde{m}_{c,k} = \text{Proj}(\tilde{m}_{c,k}^1, \tilde{m}_{c,k}^2, \ldots, \tilde{m}_{c,k}^T), \tag{10}$$

Figure 5: Qualitative Results on the CityCam dataset (Zhang et al. 2017). Comparison between the baseline (Liu et al. 2022) (top) and our proposed method (bottom) is shown. As exemplified, our method provides more robust detection results compared to the baseline.

$$\text{where } \tilde{m}^t_{c,k} = s^t_{c,k} m_{c,k} + (1 - s^t_{c,k})\varnothing, \tag{11}$$

where T is the number of sampling index $s^t_{c,k}$, and $\varnothing$ denotes learnable no-class embedding. Here $s^t_{c,k}$ is derived by binarizing $p_{c,k}$ using T thresholds τ_t (*i.e.*, $s^t_{c,k} = \delta(p_{c,k} > \tau_t)$). The delta function δ outputs 1 when the condition is true, and 0 otherwise. The projection layer Proj$(\cdot)$ combines multi-thresholded memory information with varying confidences to generate the final sampled memory.

During training, we employ asymmetric loss (Ben-Baruch et al. 2020) additionally to train the classification head and enhance the class discrimination capability of the Transformer encoder.

Memory-guided Transformer Decoder. Many recent works that have built upon DETR-like methods address the ambiguity in the role of object queries by incorporating positional information into the object queries (Meng et al. 2021; Liu et al. 2022). This has clarified the positional information of object queries, enabling them to locate objects at various positions within the current image. However, if object queries are aggregated with poor image features of the current data, they might acquire incorrect semantic information. To address this issue, we propose a method to enhance the semantic information of object queries, using a memory-attention layer. One block of our proposed memory-guided Transformer decoder (MGD) is composed of three types of attention layers, formed by adding a memory cross-attention layer to the components of the existing decoder blocks:

$$\mathcal{O}^l_{\text{sa}} = \text{Self-Attn}(Q = \mathcal{O}^l, K = \mathcal{O}^l, V = \mathcal{O}^l), \tag{12}$$

$$\mathcal{O}^l_{\text{ca}} = \text{Cross-Attn}(Q = \mathcal{O}^l_{\text{sa}}, K = \mathcal{F}, V = \mathcal{F}), \tag{13}$$

$$\mathcal{O}^{l+1} = \text{Mem.Cross-Attn}(Q = \mathcal{O}^l_{\text{ca}}, K = \tilde{M}, V = \tilde{M}), \tag{14}$$

where $\mathcal{O}$ means object queries. In the MGD, $\mathcal{O}$ acquires semantic information about the current image from the existing attention layers, and then enhances its associated class information through the memory cross-attention layer. Finally, each output object query of MGD is transformed by an FFN to output a class score and box location for each object. The subsequent processes, such as Hungarian matching and losses, follow DETR (Carion et al. 2020).

Test-time Memory Adaptation

Given the non-parametric design of our CMM, adapting our memory to test data becomes achievable without the need for additional fine-tuning. To facilitate this, we introduce a test-time adaptation strategy utilizing CMM. Within this framework, the CMM preserves representations acquired during training and independently stores contextual information specific to the target domain. To mitigate the storage of initial noisy representations, uniform coefficients are employed for the momentum update of individual memories during the training stage. During testing, on the other hand, we update the test memory module M' by individually adjusting the update rate of each memory $m'_{c,k}$ to align with the memory distribution in the test domain.

$$m'_{c,k} \leftarrow \frac{1}{i_{c,k} + J}\,(i_{c,k} m'_{c,k} + \sum_{j=1}^{J} f_j), \tag{15}$$

where $i_{c,k}$ indicates the total number of detected instances before current frame, and J is the number of instance features f_j at current frame. The update process involves adding to memory the instance features in the current image that are highly correlated with the mean values in test and source memory. For memory retrieval, a weighted sum is applied to combine memory from training and memory originating from the target domain:

$$M' \leftarrow \beta M + (1 - \beta) M', \tag{16}$$

where $\beta \in [0, 1]$ denotes the weight of the source domain. This adaptation technique ensures that memory is adapted to the target domain while preserving important contextual information from the source distribution. We can also tailor the memory to specific individual cameras, resulting in a more robust test memory module for various camera systems.

Experiments

Experimental Setup

Datasets. In this study, we assess the effectiveness of our framework through experiments conducted on two distinct datasets: the CityCam dataset (Zhang et al. 2017) and the ImageNet VID dataset (Russakovsky et al. 2015). The CityCam dataset consists of approximately 60K labeled frames, with 900K annotated objects across 10 vehicle classes. It is composed of 16 camera locations in downtown and parkway areas, spanning four typical weather conditions and time periods. For our experiments, We use 13 camera locations for training and 3 camera locations for testing. ImageNet VID consists of 3,862 training videos and 555 validation videos across 30 object classes. Following common settings in previous works (Zhou et al. 2022; Chen et al. 2020), we train CETR on the training split of ImageNet VID and DET datasets.

Model	FPS ↑	Mem ↓ (GB)	AP	AP_{50}	AP_S	AP_M	AP_L
Faster R-CNN (Girshick 2015)	37.8	0.42	23.3	39.6	18.5	39.0	36.3
Conditional DETR (Meng et al. 2021)	36.7	0.46	23.0	41.9	17.7	38.9	43.2
Conditional DETR + CETR	33.8	0.48	24.4	42.4	19.3	40.0	45.0
DAB-DETR (Liu et al. 2022)	33.5	0.47	23.8	40.2	18.6	39.4	43.4
DAB-DETR + CETR	30.7	0.48	25.0	43.0	19.7	40.4	48.4
Deformable DETR (Zhu et al. 2020)	37.9	0.42	24.2	41.0	20.1	41.4	48.0
Deformable DETR + TransVOD (Zhou et al. 2022)	4.3	4.19	23.6	40.2	19.8	40.2	47.4
Deformable DETR + CETR	30.6	0.48	**25.7**	**43.0**	**20.5**	**41.8**	**53.6**

Table 1: Quantitative results on the CityCam dataset (Zhang et al. 2017).

Implementation details. Our framework is trained on 24GB RTX-3090 GPUs with a batch size of 16, using the AdamW (Loshchilov and Hutter 2017) optimizer. We train CETR for 150K iterations, with a learning rate of 10^{-4} for the first 120K iterations and 10^{-5} for the last 30K iterations. For fast convergence, we employed variants of DETR as our baseline. In the CityCam experimentation, ResNet-50 (He et al. 2016) is used as the backbone and initialized with pre-trained weights from ImageNet dataset (Russakovsky et al. 2015), while the Transformer encoder and decoder were initialized randomly. In the ImageNet VID experiment, ResNet-101 is used as the backbone and the entire network is initialized with pre-trained weights from COCO dataset (Lin et al. 2014). For the ImageNet VID experiment and the ablation study, we used DAB-DETR with CETR.

Evaluation metric. We follow the standard COCO evaluation. We report the average precision under different IoU thresholds (AP), AP scores at IoU thresholds are 0.5 (AP_{50}), and different object scales (AP_S, AP_M, AP_L). For the ImageNet VID dataset, we follow the common protocol (Zhou et al. 2022; Chen et al. 2020; Deng et al. 2019b) and leverage average precision at IoU thresholds are 0.5 (AP_{50}) as the evaluation metric.

Experimental Results

Quantitative results. Table 1 presents our main results on the CityCam testing set, which we have divided. We applied our proposed framework to the single frame DETR-like methods and conducted quantitative comparisons with other single frame detection methods and a multi-frame DETR-like method, TransVOD (Zhou et al. 2022) that use Deformable DETR as their baseline. Compared to the single frame baseline (Zhu et al. 2020), our method showed improvements of 1.5% AP and 2.0% AP_{50}, with only a marginal increase of 0.06 GB in allocated memory and a decrease of 7.3 FPS, while the multi-frame DETR-like method showed an increase of 3.77 GB in allocated memory and a decrease of 33.6 FPS. Additionally, the multi-frame method that has mainly been utilized with video clip data of consistent short-frame intervals exhibits poor performance on the CityCam dataset with wider frame intervals. We also compare our approach with SELSA (Wu et al. 2019), LRTR (Shvets, Liu, and Berg 2019), RDN (Deng et al. 2019b), TransVOD Lite (Zhou et al. 2022), DFF (Zhu et al. 2017b), D&T (Feichtenhofer, Pinz, and Zisserman 2017), LWDN (Jiang et al. 2019), OGEMN (Deng et al. 2019a), PSLA (Guo et al. 2019), and LSTS (Jiang et al. 2020) on ImageNet VID dataset. As shown in Table 2, our method demonstrates competitive performance with other VOD methods without the need for multi-frame approaches, which require significant allocated memory and disrupt general online inference. Among online methods, it achieves the highest AP_{50} of 79.6%. Furthermore, our method exhibits higher FPS compared to most approaches, except for the method that requires multi-frame image inference at once.

Model	Online	AP_{50}	FPS ↑	#Params ↓ (M)	Mem ↓ (GB)
SELSA	✗	80.3	7.2	-	-
LRTR	✗	80.6	10	-	-
RDN	✗	**81.8**	10.6	-	-
TransVOD	✗	80.5	32.3	74.2	2.94
DFF	✓	73.1	20.25	97.8	-
D&T	✓	75.8	7.8	-	-
LWDN	✓	76.3	20	77.5	-
OGEMN	✓	76.8	14.9	-	-
PSLA	✓	77.1	18.7	63.7	-
LSTS	✓	77.2	23.0	64.5	-
CETR	✓	**79.6**	23.3	65.7	0.55

Table 2: Performance comparison with state-of-the-art real-time VOD methods with ResNet-101 backbone on ImageNet VID dataset (Russakovsky et al. 2015). Here we use AP_{50}, which is commonly used as mean average precision (mAP) in other VOD methods.

Qualitative results. As shown in Fig. 4, we provide a visualization of the correlation between the proposed memory and image features. To do this, we utilize the attention map in the 4th layer of the Transform encoder. The CMM exhibits notable attention scores for objects of the corresponding class. This observation emphasizes that class-specific memory clearly contains the relevant information specific

Methods	CMM	MGD	SS	MT	AP_{50}
Baseline	-	-	-	-	40.2
Ours	✓	-	-	-	40.5
	✓	✓	-	-	40.5
	✓	✓	✓	-	41.1
	✓	✓	✓	✓	**42.5**

Table 3: Ablation study on main components. CMM, SS, MT, and MGD denote Context Memory Module, score-based sampling, Multi-level thresholding, and Memory-Guided Transformer Decoder, respectively.

# Prototype (K)	AP	AP_{50}	AP_S	AP_M	AP_L
1	23.7	41.9	18.2	38.9	46.6
3	**24.8**	**42.5**	**19.6**	**40.0**	**47.4**
5	23.2	41.1	17.4	38.8	45.1
10	23.3	40.0	18.2	39.1	45.6

Table 4: Performance for the number of prototypes per class.

to each class. In addition, we visually compare the object detection results of the baseline model and our approach in Fig. 5. The results showcase higher confidence scores for most classes in our approach compared to the baseline. Notably, comparing the 2nd and 4th columns, we notice that our model performs well in detecting objects that are partially visible within the frame, while the baseline fails. In addition, the first qualitative result shows the superiority of our model in capturing even rare classes (e.g., small trucks) within the dataset.

Ablation Study and Analysis

Memory module analysis. We first investigate the effect of our main components. As shown in Table 3, when our CMM is used only in the Transformer encoder, it has a 0.3% improvement AP_{50} compared to the single frame baseline (Liu et al. 2022). When used in isolation, the MGD does not exhibit any performance enhancement. However, when used in combination with the score-based sampling method, there was an increase of 0.6% AP_{50}. In addition, when used in conjunction with the multi-level thresholding method, it shows an additional improvement of 1.4% AP_{50}.

The number of prototypes. Table 4 illustrates the ablation study on the number of prototypes of each class in our context memory module. When using one prototype and three prototypes, our approach achieves performance improvements of 1.7% and 2.3% AP_{50} compared to the single frame baseline, respectively. However, we observe that the performance of our approach in the CityCam dataset decreases as the number of prototypes is increased beyond 3. This outcome is believed to be due to the CityCam dataset consisting solely of classes grouped under the vehicle category that share similar types. As a result, a small number of prototypes is enough to represent the distribution of class features, and too many prototypes may, in fact, hinder the utilization of class features.

Method	AP_{50}
Baseline	40.2
Learnable memory	40.9
Full memory	40.5
Random sampling	41.2
Score-based sampling	**42.5**
GT sampling (oracle)	57.6

Table 5: Experiments on sampling strategy.

Method	AP	AP_{50}	AP_S	AP_M	AP_L
Our Baseline	24.8	42.5	19.6	40.0	47.4
+ Memory update	24.9	42.8	**19.7**	40.2	47.8
+ Cam specific	**25.0**	**43.0**	**19.7**	**40.4**	**48.4**

Table 6: Experiments on the test-time memory adaptation.

Sampling strategy. Table 5 reports the performance of our approach according to various class-wise memory sampling strategies for the memory-guided Transformer decoder. When employing the class-wise memory sampling strategy using ground-truth images, If we have an oracle-level knowledge of the correct answers, it shows a significant performance improvement of 17.4% AP_{50}. Taking this result as motivation, we designed our score-based sampling module. The performance of the approaches improves in each case when using all learnable memory or randomly sampling class-wise memory. However, the classification score-based sampling strategy leads to the most improvement in performance. This result demonstrates the effectiveness of our score-based sampling method.

Test-time memory adaptation. Lastly, we conduct extensive experiments to assess the effectiveness of the CMM-based test-time adaptation approach, as shown in Table 6. The results highlight that the memory update technique adapted to the target domain yields meaningful performance improvements without requiring further training or parameter optimization. From the table, we also notice that by adding a camera-specific way of configuring memory, performance can be improved by 0.5% for the AP_{50} over the baseline.

Conclusion

To handle video data with a single frame approach, we introduced a context memory module that enables the use of spatio-temporal contextual information from the entire dataset. In addition, we used a score-based sampling and a memory-guided transformer decoder to effectively make use of our context memory. Our method exhibited a meaningful performance improvement over the single frame baseline in the CityCam dataset, with only a slight increase in allocated memory and a low decrease in FPS. Furthermore, our method demonstrated a remarkable performance improvement over other real-time online video object detection methods when evaluated on the ImageNet VID dataset.

Acknowledgments

This research was supported by the MSIT, Korea (IITP-2023-2020-0-01819, RS-2023-00222280), National Research Foundation of Korea (NRF-2021R1C1C1006897, NRF-2018M3E3A1057288).

References

Ben-Baruch, E.; Ridnik, T.; Zamir, N.; Noy, A.; Friedman, I.; Protter, M.; and Zelnik-Manor, L. 2020. Asymmetric loss for multi-label classification. *arXiv preprint arXiv:2009.14119*.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, 213–229. Springer.

Chen, C.; Seff, A.; Kornhauser, A.; and Xiao, J. 2015. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE international conference on computer vision*, 2722–2730.

Chen, D.; Wang, D.; Darrell, T.; and Ebrahimi, S. 2022. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 295–305.

Chen, X.; Kundu, K.; Zhang, Z.; Ma, H.; Fidler, S.; and Urtasun, R. 2016. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2147–2156.

Chen, Y.; Cao, Y.; Hu, H.; and Wang, L. 2020. Memory enhanced global-local aggregation for video object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10337–10346.

Chen, Y.; Xu, X.; Su, Y.; and Jia, K. 2023. STFAR: Improving Object Detection Robustness at Test-Time by Self-Training with Feature Alignment Regularization. *arXiv preprint arXiv:2303.17937*.

Cui, Y. 2023. Feature Aggregated Queries for Transformer-Based Video Object Detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6365–6376.

Dai, J.; Li, Y.; He, K.; and Sun, J. 2016. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, 29.

Deng, H.; Hua, Y.; Song, T.; Zhang, Z.; Xue, Z.; Ma, R.; Robertson, N.; and Guan, H. 2019a. Object guided external memory network for video object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6678–6687.

Deng, J.; Pan, Y.; Yao, T.; Zhou, W.; Li, H.; and Mei, T. 2019b. Relation distillation networks for video object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 7023–7032.

Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; and Brox, T. 2015. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 2758–2766.

Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; and Tian, Q. 2019. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6569–6578.

Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2017. Detect to track and track to detect. In *Proceedings of the IEEE international conference on computer vision*, 3038–3046.

Fu, Z.; Chen, Y.; Yong, H.; Jiang, R.; Zhang, L.; and Hua, X.-S. 2019. Foreground gating and background refining network for surveillance object detection. *IEEE Transactions on Image Processing*, 28(12): 6077–6090.

Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Guo, C.; Fan, B.; Gu, J.; Zhang, Q.; Xiang, S.; Prinet, V.; and Pan, C. 2019. Progressive sparse local attention for video object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3909–3918.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hernández, A. C.; Gómez, C.; Crespo, J.; and Barber, R. 2016. Object detection applied to indoor environments for mobile robot navigation. *Sensors*, 16(8): 1180.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Iwasawa, Y.; and Matsuo, Y. 2021. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34: 2427–2440.

Jang, M.; and Chung, S.-Y. 2022. Test-time adaptation via self-training with nearest neighbor information. *arXiv preprint arXiv:2207.10792*.

Jiang, Z.; Gao, P.; Guo, C.; Zhang, Q.; Xiang, S.; and Pan, C. 2019. Video Object Detection with Locally-Weighted Deformable Neighbors. In *AAAI Conference on Artificial Intelligence*.

Jiang, Z.; Liu, Y.; Yang, C.; Liu, J.; Gao, P.; Zhang, Q.; Xiang, S.; and Pan, C. 2020. Learning where to focus for efficient video object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, 18–34. Springer.

Kang, K.; Li, H.; Xiao, T.; Ouyang, W.; Yan, J.; Liu, X.; and Wang, X. 2017a. Object detection in videos with tubelet proposal networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 727–735.

Kang, K.; Li, H.; Yan, J.; Zeng, X.; Yang, B.; Xiao, T.; Zhang, C.; Wang, Z.; Wang, R.; Wang, X.; et al. 2017b. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10): 2896–2907.

Li, F.; Zhang, H.; Liu, S.; Guo, J.; Ni, L. M.; and Zhang, L. 2022. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13619–13627.

Li, Y.; Wang, N.; Shi, J.; Liu, J.; and Hou, X. 2016. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Liu, S.; Li, F.; Zhang, H.; Yang, X.; Qi, X.; Su, H.; Zhu, J.; and Zhang, L. 2022. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Meng, D.; Chen, X.; Fan, Z.; Zeng, G.; Li, H.; Yuan, Y.; Sun, L.; and Wang, J. 2021. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3651–3660.

Nascimento, J. C.; and Marques, J. S. 2006. Performance evaluation of object detection algorithms for video surveillance. *IEEE Transactions on Multimedia*, 8(4): 761–774.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252.

Shvets, M.; Liu, W.; and Berg, A. C. 2019. Leveraging long-range temporal relationships between proposals for video object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9756–9764.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9627–9636.

Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B.; and Darrell, T. 2020. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*.

Wang, H.; Tang, J.; Liu, X.; Guan, S.; Xie, R.; and Song, L. 2022a. Ptseformer: Progressive temporal-spatial enhanced transformer towards video object detection. In *European Conference on Computer Vision*, 732–747. Springer.

Wang, Q.; Fink, O.; Van Gool, L.; and Dai, D. 2022b. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7201–7211.

Wang, S.; Zhou, Y.; Yan, J.; and Deng, Z. 2018. Fully motion-aware network for video object detection. In *Proceedings of the European conference on computer vision (ECCV)*, 542–557.

Wu, H.; Chen, Y.; Wang, N.; and Zhang, Z. 2019. Sequence level semantics aggregation for video object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9217–9225.

Zhang, S.; Wu, G.; Costeira, J. P.; and Moura, J. M. 2017. Understanding traffic density from large-scale web camera data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5898–5907.

Zhou, Q.; Li, X.; He, L.; Yang, Y.; Cheng, G.; Tong, Y.; Ma, L.; and Tao, D. 2022. TransVOD: end-to-end video object detection with spatial-temporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhu, X.; Dai, J.; Yuan, L.; and Wei, Y. 2018. Towards high performance video object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7210–7218.

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

Zhu, X.; Wang, Y.; Dai, J.; Yuan, L.; and Wei, Y. 2017a. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE international conference on computer vision*, 408–417.

Zhu, X.; Xiong, Y.; Dai, J.; Yuan, L.; and Wei, Y. 2017b. Deep feature flow for video recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2349–2358.