mathematics

MDPI

Article

Modulated Memory Network for Video Object Segmentation

Hannan Lu, Zixian Guo and Wangmeng Zuo *

Faculty of Computing, Harbin Institute of Technology, Harbin 150001, China; hannanlu@hit.edu.cn (H.L.); 23B903063@stu.hit.edu.cn (Z.G.)
* Correspondence: wmzuo@hit.edu.cn

Abstract: Existing video object segmentation (VOS) methods based on matching techniques commonly employ a reference set comprising historical segmented frames, referred to as 'memory frames', to facilitate the segmentation process. However, these methods suffer from the following limitations: (*i*) Inherent segmentation errors in memory frames can propagate and accumulate errors when utilized as templates for subsequent segmentation. (*ii*) The non-local matching technique employed in top-leading solutions often fails to incorporate positional information, potentially leading to incorrect matching. In this paper, we introduce the Modulated Memory Network (MMN) for VOS. Our MMN enhances matching-based VOS methods in the following ways: (*i*) Introducing an Importance Modulator, which adjusts memory frames using adaptive weight maps generated based on the segmentation confidence associated with each frame. (*ii*) Incorporating a Position Modulator that encodes spatial and temporal positional information for both memory frames and the current frame. The proposed modulator improves matching accuracy by embedding positional information. Meanwhile, the Importance Modulator mitigates error propagation and accumulation by incorporating confidence-based modulation. Through extensive experimentation, we demonstrate the effectiveness of our proposed MMN, which also achieves promising performance on VOS benchmarks.

Keywords: video object segmentation; matching based; memory network; modulator

MSC: 68T45

Citation: Lu, H.; Guo, Z.; Zuo, W. Modulated Memory Network for Video Object Segmentation. *Mathematics* **2024**, *12*, 863. https://doi.org/10.3390/math12060863

Academic Editor: Ivan Lorencin

Received: 16 February 2024
Revised: 2 March 2024
Accepted: 11 March 2024
Published: 15 March 2024

1. Introduction

Video object segmentation (VOS) is an important research area within the domain of video analysis, with the potential to yield significant advantages across various computer vision applications, including but not limited to video editing and autonomous driving. The primary focus of this research is one-shot VOS, a task aimed at the continuous segmentation of specific object(s) throughout an entire video sequence, with pixel-wise mask annotations for the object(s) of the first frame provided.

Recent developments in VOS techniques have explored template matching methods, using a reference set of historical frames and their corresponding object masks. This reference set serves as the foundation for propagating labels to subsequent frames through pixel-wise matching. Notably, by introducing memory networks [1,2] within the Spatiotemporal Memory Network (STM) framework [3], a spatiotemporal memory is initially constructed for segmentation, allowing frames with intermediate predictions to be periodically updated to memory. The majority of current top-performing one-shot VOS methods [4–10] are based on the STM architecture, exploiting its ability to utilize comprehensive reference information from distant temporal frames. However, these matching-based methods still suffer from two significant challenges: (*i*) Inherent errors present in intermediate predictions may result in error propagation and accumulation when utilizing segmented frames as references. (*ii*) The widely employed non-local matching technique, prevalent in the aforementioned approaches, lacks the incorporation of positional information, potentially leading to incorrect matching.

In this study, we introduce the Modulated Memory Network (MMN), integrating two key components, namely, the Importance Modulator (IM) and the Position Modulator (PM), which are designed to tackle the aforementioned challenges. As illustrated in Figure 1, the IM module performs pixel-level modulation on the features of historical frames within the memory, utilizing adaptive weights that operate across both spatial and channel dimensions. These modulation weights are generated based on the retrieval from memory and the segmentation confidence associated with historical frames, serving as crucial conditions. Through adaptive feature modulation, incorrectly segmented pixels within the memory exert less influence on subsequent segmentation, effectively mitigating the propagation and accumulation of errors. Simultaneously, the PM module incorporates positional embedding into the features of both query and memory frames, encoding spatiotemporal coordinates. This positional embedding enhances the robustness of memory retrieval in the presence of distractions. Furthermore, among all historical frames, the initial frame and the last frame hold greater significance. The initial frame carries ground-truth annotations, while the last frame provides substantial spatial guidance. A toy experiment, as displayed in Table 1, demonstrates a more pronounced performance drop when removing these two frames from memory, as opposed to removing other historical frames. To optimize the utilization of the initial and the most recent frames, their temporal positions are given a special encoding.

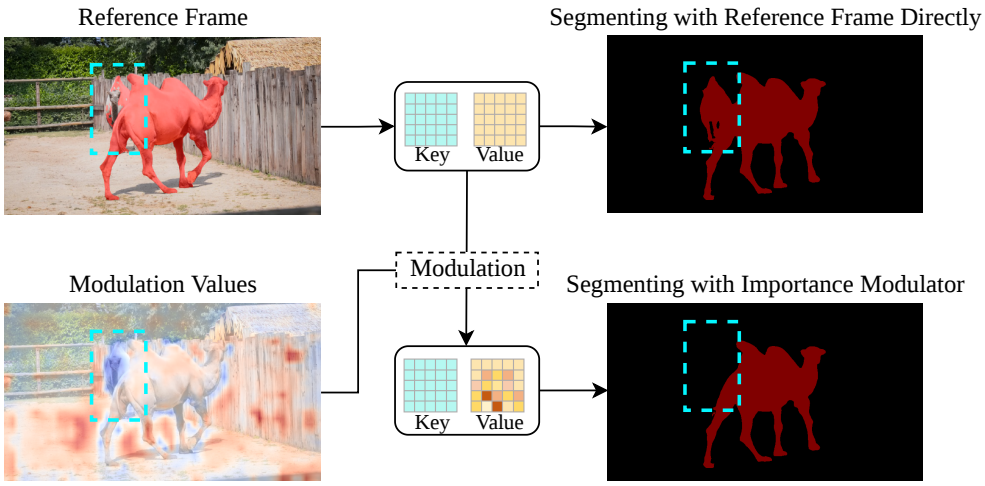

Figure 1. A comparison of segmentation results when utilizing previously computed frames as a reference. The dashed bounding box highlights inaccurately segmented pixels in the reference frame, which introduces cumulative errors in subsequent frames. Conversely, our IM module effectively mitigates error propagation by modulating the inaccurately segmented pixels with reduced values (depicted in blue).

Table 1. Quantitative results comparing the importance of frames at different temporal positions to matching-based VOS methods. Two state-of-the-art matching-based solutions STM [3] and AFB-URR [7] are chosen for comparison. The 'Sample Rate' is the proportion of intermediate frames collected to memory. For these two methods, without the first or last frame, performance degrades more significantly than that without multiple intermediate frames. The $\mathcal{J}\&\mathcal{F}$ score is the mean of region similarity $\mathcal{J}$ and contour accuracy $\mathcal{F}$, from DAVIS [11].

First/Last	Sample Rate	$\mathcal{J}\&\mathcal{F}$	
		STM [3]	**AFB-URR [7]**
-	1/5	81.7	74.6
w/o first	1/5	80.1 (−1.6)	73.5 (−1.1)
w/o last	1/5	80.1 (−1.6)	74.0 (−0.6)
-	1/15	81.5 (−0.2)	74.6 (−0.0)

The primary contributions of this study can be summarized as follows:

1. We introduce the Modulated Memory Network (MMN) for video object segmentation (VOS), incorporating both an Importance Modulator (IM) and a Position Modulator (PM). The MMN, as proposed, conducts adaptive memory modulation that takes into account segmentation confidence and spatiotemporal positional information. This enhancement results in improved object segmentation accuracy.
2. The IM module, designed to execute pixel-level modulation, utilizes adaptive weights generated based on segmentation confidence as conditions. Simultaneously, the PM module is implemented to embed positional information into both query and memory frames, thereby enhancing matching accuracy.
3. Our proposed Modulated Memory Network (MMN) consistently achieves performance comparable to state-of-the-art methods for established VOS benchmarks. Furthermore, ablation studies provide additional evidence of the effectiveness of the IM and PM modules.

2. Related Work

According to task specifications, the video object segmentation (VOS) task can be categorized into zero-shot VOS, one-shot VOS and interactive VOS. Zero-shot VOS [12–26] involves fully automated segmentation of salient objects in a video without any manual initialization. One-shot VOS [3–7,9,10,27–41] focuses on segmenting a specified object, typically provided in the first frame. Interactive VOS [42–47] incorporates manual interaction steps. This paper addresses the relatively foundational one-shot VOS task.

Deep learning-based methods for one-shot VOS can be categorized into three distinct groups, based on their utilization of the first annotated frame during inference, as outlined by Wang et al. [48]. These categories include online fine-tuning-based methods, motion-based methods and matching-based methods. Online fine-tuning-based methods, exemplified by works such as OSVOS [29], OSVOS-S [30] and OnAVOS [38], based on the idea of image segmentation [49–51], involve the fine-tuning of general segmentation networks on the initial annotated frame to target specific objects. In order to address the evolving appearance of objects over time, these methods frequently conduct online updates of their segmentation models. However, this online training incurs a notable increase in testing time. In contrast, motion-based approaches, represented by works like MaskRNN [52], MaskTrack [53] and PReMVOS [39], leverage temporal continuity. These methods estimate pixel-wise motion to transform object masks from the first annotated frame to subsequent frames. Although motion-based techniques offer computational efficiency, they are susceptible to object drifts caused by occlusions and rapid object movements.

Matching-based one-shot VOS: In recent times, matching-based methods have gained prominence in video tasks [54] due to their resilience in handling occlusions and rapid object motion. In the one-shot VOS task, this category of approaches formulates the task as a template matching problem, initializing a reference set with historical frames and their respective masks. Subsequently, labels are propagated from this reference set to the current frame (the query frame) through pixel-wise matching and retrieval.

Early efforts [37,55] incorporated siamese structure networks to extract features from both the reference and query frames. Notably, VideoMatch [37] introduced the practice of updating the reference set with segmented frames. Meanwhile, OSMN [28] introduced modulators to adapt a generic segmentation network to specific objects without resorting to online fine-tuning. Under the assumption of temporal continuity, FEELVOS [35] narrowed the matching area using the last frame as a local range. Building on FEELVOS [35], CFBI [33] further enhanced the approach by establishing separate reference sets for the foreground and background.

Leveraging memory networks [1,2,56], STM [3] constructed an extensive memory encompassing pixel-level features from the reference set. The success of STM [3] significantly influenced the development of matching-based solutions. EGMN [6] introduced a graph-structured memory network, performing memory reading and writing on each node

sequentially. To enhance inference efficiency, dynamic memory update strategies were explored. For instance, GCM [4] maintained a fixed memory size, updating each computed result to memory with equal weight. SwiftNet [8] triggered memory updates based on cumulative differences across frames. In order to remove obsolete features, AFB-URR [7] maintained a least-frequently used index for each feature. However, the importance of each reference feature needed to be distinguished before use, as they originated from pixels with varying levels of segmentation confidence. In our work, we propose the Modulated Memory Network, where an Importance Modulator is incorporated for reference feature modulation.

To mitigate the ambiguity of similar objects, techniques were devised to impose constraints on the retrieval of reference information. KMN [5], for instance, enhances memory reading operations with Gaussian kernels to reduce non-localization issues. RMNet [10] restricts matching to specific regions by employing bounding boxes around objects. Meanwhile, LCM [9] reinforces potential object regions with an additional position guidance module. However, these approaches do not fully harness spatial and temporal positional information nor do they explore their interconnection.

Position encoding: To compensate for the limitations of permutation-invariant architectures, Transformer [57] introduced 1D positional information encoding through positional embedding. Further advancements can be observed in approaches like DETR [58] and VisTR [57], which apply positional embedding to multi-dimensional data encompassing images and videos. In these methods, positional information for each dimension is encoded separately and then concatenated. Despite these explorations, it is believed that the positional encoding technique can be refined further by capitalizing on the inherent characteristics of a one-shot VOS task.

3. Method

3.1. Overview

As shown in Figure 2, given a video sequence $\mathcal{S} = \{I_i\}_{i=1}^{T}$, where the initial frame I_1 is provided with an associated mask P_1, the objective of one-shot VOS is to segment the designated object(s) from the background by predicting corresponding masks $P = \{P_i\}_{i=2}^{T}$. Following the paradigm of matching-based VOS models, MMN processes each frame in a sequential manner.

The segmentation process can be divided into two fundamental steps, namely, memorization and segmentation, which are carried out iteratively. In the memorization step, our Modulated Memory Network (MMN) constructs a modulated memory. Within this memory, features from historical frames are extracted and subsequently modulated by the Importance Modulator (IM). In the segmentation step, the current frames retrieve supportive features from the modulated memory through pixel-wise matching. This process embeds spatiotemporal information into the features of both the current frames and the frames stored in memory, facilitated by the Position Modulator (PM). Ultimately, a decoder is employed to transform the memory retrieval into an object mask. In the subsequent subsections, we delve into the specifics of these two steps and elucidate the architecture of the proposed modules.

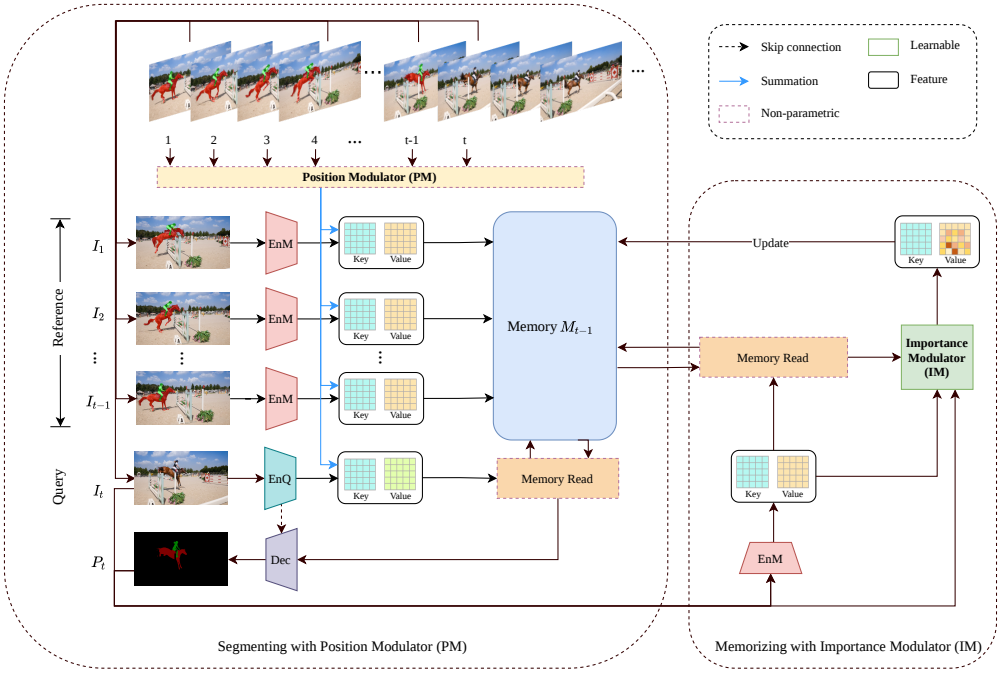

Figure 2. An overview of our proposed MMN method. The specially designed Position Modulator (PM) enhances both query and memory embeddings with spatiotemporal positional information, enabling accurate reference retrieval for segmentation. Concurrently, the Importance Modulator (IM) takes segmentation quality and contextual importance into account when performing modulation on the memory, resulting in an informative memory.

3.2. Memorization

In memorization, a modulated memory is constructed for the subsequent segmentation using historical frames and their corresponding masks. The historical frames are encoded and stored in the memory sequentially. As shown in Figure 3, for historical frame I^i and its object mask P^i, a memory encoder EnM is first applied, which takes input of both I^i and P^i and output key feature $K_M^i \in \mathbb{R}^{HW \times C_k}$ and value feature $V_M^i \in \mathbb{R}^{HW \times C_v}$. C_k and C_v are the channel dimensions of the key and value features, which are 128 and 512 in our MMN. Spatial dimensions are $H = sH_{im}$ and $W = sW_{im}$, where H_{im} and W_{im} correspond to the spatial dimension of the input frames. s corresponds to the stride of the backbone in our MMN, which is 16 since we use ResNet-50 [59] as the backbone following STM [3]. Previous solutions [3,5,10] store K_M^i and V_M^i to memory directly. However, for the inherent errors in the intermediate mask predictions, segmenting with K_M^i and V_M^i may lead to error propagation and accumulation. To address this issue, we construct a modulated memory with the proposed IM module, which modulates each historical frame, respectively, considering their segmentation reliability and context importance. In particular, the IM module takes as input the value feature V_M^i, a readout feature R_M^i and the object mask P^i, and it outputs a weight map A^i. To obtain R_M^i, the features of $\{I_j\}_{j=1}^{i-1}$, i.e., key features $K_M^{i-1} = \{K_M^j\}_{j=1}^{i-1}$ and value features $V_M^{i-1} = \{V_M^j\}_{j=1}^{i-1}$, are taken as a reference. An affinity matrix A^i is first calculated between the key features. For each spatial location p in K_M^i and q in $K_M^{i-1} = \{K_M^j\}_{j=1}^{i-1}$,

$$A(p,q) = \frac{\exp(S_{p,q})}{\sum_q \exp(S_{p,q})} \tag{1}$$

Function S calculates the similarity, which is calculated as follows:

$$S_{p,q} = \mathcal{C}(K_M^i(p), K_M^{i-1}(q)) \tag{2}$$

$\mathcal{C}$ is the dot production function as in STM [3]. Then, R_M^i is aggregated through the weighted sum of value features $V_M^{i-1} = \{V_M^j\}_{j=1}^{i-1}$,

$$R_M^i(p) = \sum_q (A(p,q) \cdot V_M^{i-1}(q)) \tag{3}$$

The IM module can learn to estimate the reliability of frame I^i by comparing R_M^i with V_M^i. Afterward, we adopt EnP to encode P^i into feature P_E^i, of which the number of channels is 32. Then, we concatenate P_E^{i-1}, R_M^i and V_M^i together along the channel dimension, and we input it into an estimator Est and output the modulation map A_i. Finally, modulation for frame I^i is conducted as follows, for each position p in V_M^i:

$$\hat{V}_M^i(p) = A(p) \cdot V_M^i(p) \tag{4}$$

The modulation is only conducted on value features, which mainly carry the label information of the target object. We provide experimental results in Section 4.3, comparing different modulation modes. After the modulation, the modulated features of the historical frames $\{I_j\}_{j=1}^i$ are stacked together along the temporal dimension, constructing a modulated memory $M^i = \{K_M \in \mathbb{R}^{NHW \times C_k}, \hat{V}_M \in \mathbb{R}^{NHW \times C_v}\}$.

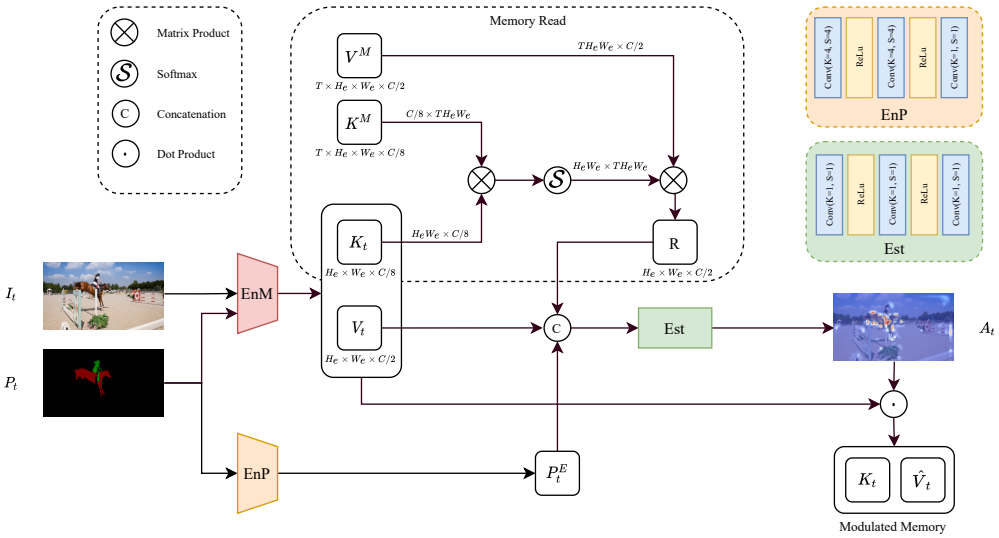

Figure 3. The architecture of the proposed Importance Modulator (IM). The IM module takes as input the concatenation of query embedding V_t, retrieved reference information R and the encoded embedding of prediction P_t^E. It then produces a modulation factor A_t, which signifies the importance of each pixel.

3.3. Segmentation

In segmentation, MMN segments the current frame I^t using the modulated memory M^{t-1}, where support features are retrieved through matching. To improve matching accuracy, the Position Modulator (PM) is deployed to encode spatiotemporal positional information for the key features of both memory frames and the current frame, which also helps to better exploit the two vital frames (i.e., the first and the last frames). Here, we adapt a positional encoding in VisTR [57] for the VOS task. In particular, coordinates in each dimension (i.e., height, width and temporal dimensions) are first normalized and encoded into embeddings using sine and cosine functions with different frequencies:

$$PE(p,c) = \begin{cases} \sin(\text{Norm}(p) \cdot \omega_k), & \text{for } c = 2k \\ \cos(\text{Norm}(p) \cdot \omega_k), & \text{for } c = 2k+1 \end{cases} \tag{5}$$

where p is the coordinate in the corresponding dimension, $\omega_k = 1/10{,}000^{2k/\frac{d}{3}}$ and c is the channel index. Norm is a normalization function. The spatial coordinates (i.e., height and width dimensions) are sequentially numbered, and their normalization function is defined as

$$\mathrm{Norm}_R(p) = \frac{p}{\max_{p \in P}((p))} \tag{6}$$

As for the temporal dimension, to better exploit the two vital frames, we specially define temporal coordinates as $P_t = \{2, 3, 4, \ldots, t-1, 1, 1\}$ for each video sequence $S = \{I_1, I_2, I_3, \ldots, I_{t-2}, I_{t-1}, I_t\}$. The normalization function for temporal coordinates is defined as

$$\mathrm{Norm}_R(p) = \frac{1}{p},\ p \in P_t \tag{7}$$

The customized encoding scheme fixes the relative positional relationship between the current frame and the two vital frames, which makes it possible to approach important reference information according to their temporal positions. Positional embeddings for each dimension are concatenated together along the channel dimension and added to the key features K_M and K_Q^t element-wise, obtaining modulated key features $\hat{K}_M$ and $\hat{K}_Q^t$. $K_Q^t \in \mathbb{R}^{HW \times C_k}$ is the key feature of the current frame I^t, encoded by EnQ.

After obtaining modulated key features, the affinity matrix can be calculated the same as Equation (1). Then, support features are retrieved from the modulated memory through weighted aggregation as in Equation (4). Finally, the mask prediction of frame I^t is generated through decoding the retrieved support features by the decoder Dec.

3.4. Network Architecture and Loss Function

Same as the baseline work [3], both encoders EnM and EnQ use ResNet-50 [59] as backbones, and the output of *res-4* is used as a feature map. The parameters of EnM and EnQ are not shared. Two convolution layers are adopted to transform the output of layer *res-4* into key and value features. The decoder Dec shares the same architecture as that of STM [3], which is composed of several residual blocks and bilinear interpolation layers for upsampling.

The IM module has two learnable sub-modules, i.e., encoder EnP and estimator Est. The encoder EnP consists of 3 convolution layers. The first two are 4×4 convolution layers with stride 4, and the number of channels is 64. The last is a 1×1 convolution layer with stride 1, and the number of channels is 32. The output resolution is $\frac{1}{16}$ of the input mask resolution. The Est is constructed with three 1×1 convolution layers with stride 1. The channel numbers of these three convolution layers are 1024, 512 and 1, respectively. Both EnP and Est activate the first two layers with ReLU [60].

We train our MMN with cross-entropy loss and mask IoU loss [61], where the mask IoU loss is defined as

$$\mathcal{L}_{mIoU}(P_i, G_i) = 1 - \frac{\sum_{p \in \Omega} \min(P_i^p, G_i^p)}{\sum_{p \in \Omega} \max(P_i^p, G_i^p)} \tag{8}$$

where P and G are the predicted mask and ground-truth mask of object i and Ω represents all pixels in masks P and G.

3.5. Algorithm

The workflow of the proposed MMN is illustrated in Algorithm 1. The algorithm takes as input a sequence of video frames $\{I_1, I_2, \ldots, I_T\}$ and a given ground-truth mask P_1. The algorithm initializes memory using the first frame I_1 and the given mask P_1 and then alternates between the segmentation and memorization processes until it obtains segmentation results $\{P_2, \ldots, P_T\}$ for frames $\{I_2, \ldots, I_T\}$.

Algorithm 1 Pseudo-code for Modulated Memory Network

Input: Frames $\{I_1, I_2, \ldots, I_T\}$, ground-truth mask P_1.
Output: Predicted mask $\{P_2, \ldots, P_T\}$.

▷ Memory initialization
$K_1^M, V_1^M = \text{EnK}((I_1, P_1))$
$K^M = \{K_1^M\}, V^M = \{V_1^M\}$
for $i = 1 : T$ **do**
▷ Segmentation
$K^Q, V^Q = \text{EnQ}(I_i)$
▷ Position modulation
$PE_{spatial} = \text{PosEncoding}(H_{K^M}, W_{K^M})$ # shape=$(H, W, 1)$
$PE_{temporal} = \frac{1}{\text{range}(\text{length}(M)+1)}$ # shape=$(1, 1, i+1)$
$PE_{spatial} = \text{expand}(PE_{spatial}, (H, W, i+1))$
$PE_{temporal} = \text{expand}(PE_{temporal}, (H, W, i+1))$
$PE = \text{Concat}([PE_{spatial}, PE_{temporal}])$
$K^M = K^M + PE_{[:-1]}$
$K^Q = K^Q + PE_{[-1]}$
▷ Memory retrieval
$R^Q = (K^{Q'} \otimes K^M)V^M$
▷ Decode
$P_i = \text{Dec}(R^Q, V^Q)$
▷ Memorization
▷ Weight map generation
$K_i^M, V_i^M = \text{EnK}((I_i, P_i))$
$P_i^E = \text{EnP}(P_i)$
$R = (K_i^M, K^M)V^M$
$A_i = \text{Est}([V_i^M, P_i^E, R])$
▷ Memory modulation
$V_i^M = V_i^M \cdot A_i$
▷ Memory update
$K^M \leftarrow K^M \cup \{K_i^M\}$
$V^M \leftarrow V^M \cup \{V_i^M\}$
end for

4. Experiments

4.1. Datasets and Evaluation

Consistent with common practices, we utilize the DAVIS 2016 [62], DAVIS 2017 [11], YouTube-VOS 2018 [63] and MOSE [64] datasets for training and evaluating the proposed MMN. The DAVIS datasets [11,62] provide high-quality object annotations. The 2016 version [62] comprises 30 videos for training and 20 for validation. Each video features a single annotated object instance in its initial frame. On the other hand, the 2017 version [11] serves as a multi-object extension, incorporating a larger number of videos in both training and validation splits. Additionally, the YouTube-VOS 2018 [63] dataset represents a formidable and expansive challenge in video object segmentation. The training set contains 3471 videos, while the validation set includes 474 videos. Among the videos in the validation set, 65 semantic categories overlap with those in the training set, forming the *seen*

subset. In contrast, the remaining 26 semantic categories in the validation set differ from those in the training set, constituting the *unseen* subset. The MOSE [64] dataset is designed for complex scenes, which includes 1507 videos for training and 311 videos for validation. MOSE covers various challenging scenarios, such as interactions between multiple objects, occlusions and background interference.

Our evaluation of the MMN is conducted on the validation splits of all datasets. For DAVIS 2016 and 2017, we employ the standard evaluation toolkit provided by DAVIS 2017 [11]. This toolkit computes the mean of region similarity $\mathcal{J}$, contour accuracy $\mathcal{F}$ and their average $\mathcal{J}\&\mathcal{F}$ for comparative analysis. In the case of YouTube-VOS 2018, we report $\mathcal{J}$ and $\mathcal{F}$ metrics separately for both the *seen* and *unseen* semantic categories, as well as the overall averaged score $\mathcal{G}$. Metrics for YouTube-VOS and MOSE are all calculated through the competition server on Codalab.

4.2. Training and Inference

Following SwiftNet [8], we commence by pre-training our MMN on synthetic data [65], followed by fine-tuning using video data. During the pre-training stage, random affine transformations with varied parameters are applied to static images in order to generate synthetic video clips. We also employ a Cut and Paste strategy for augmenting images containing fewer than three target objects. This strategy involves cutting object instances from source images and pasting them into random positions within target images. The pre-training phase consists of a total of 6×10^5 iterations, employing a constant learning rate of 1×10^{-5}. The backbones of EnQ and EnM are initialized with weights pre-trained on ImageNet [66].

Upon completing the pre-training stage, the MMN undergoes fine-tuning on video datasets, with each training batch containing three frames randomly sampled from a video sequence. The sampling interval progressively increases from 1 to 25 during the first 2.5×10^4 iterations and subsequently decreases to 5 in the final 2×10^5 iterations. Random affine transformations, with distinct parameters for each frame, are applied throughout this process. The input patch size is set to 384×384, and a batch size of 4 is achieved through gradient accumulation. The fine-tuning stage comprises a total of 8×10^5 iterations and employs a poly learning rate strategy. We use Adam [67] optimization with standard momentum and no weight decay for both the pre-training and fine-tuning stages. The MMN is trained on a Tesla V100 GPU and evaluated on a GTX 2080Ti GPU. GPUs are both manufactured by NVIDIA, located in Santa Clara, CA, USA.

4.3. Ablation Study

To demonstrate the effectiveness of each component of the proposed MMN, we conduct ablation experiments on the DAVIS 2017 validation set. For fast verification, models for ablation experiments are trained directly with the DAVIS and YouTube-VOS train set, without the pre-training stage.

Importance Modulator: The IM module is devised to modulate memory frames while considering their contextual importance and segmentation confidence. To assess the effectiveness of the proposed IM module, we compare models with and without the IM module. Furthermore, we explore various modulation approaches by comparing models where modulation is applied to the key features, value features and both key and value features. As depicted in Table 2, applying modulation to the value features yields the most favorable performance. This is attributed to the value feature primarily encoding information related to predicted masks, where modulation serves to prevent the propagation and accumulation of inherent errors. In contrast, the key feature primarily encodes appearance information used for similarity calculations. Modulating the key feature could potentially lead to incorrect object matching, thereby diminishing segmentation accuracy.

We also present qualitative results in Figure 4. It is evident that, in the absence of the IM module, segmentation errors within the memory frame (highlighted in the red box) propagate to subsequent frames (*21* and *22*). However, with the utilization of the IM

module, adaptive modulation is applied to the inaccurately segmented region, effectively halting the propagation of errors.

Table 2. The results of ablation studies. We conduct an ablation study on the proposed Importance Modulator (IM). ✓ and ✗ indicate whether the IM module performs modulation on the key or value embedding. Bold indicates the **best** performance.

Key	Value	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
✗	✗	83.5	81.3	85.9
✓	✗	78.1	75.7	80.8
✗	✓	**85.8**	**82.8**	**88.9**
✓	✓	84.7	82.4	87.1

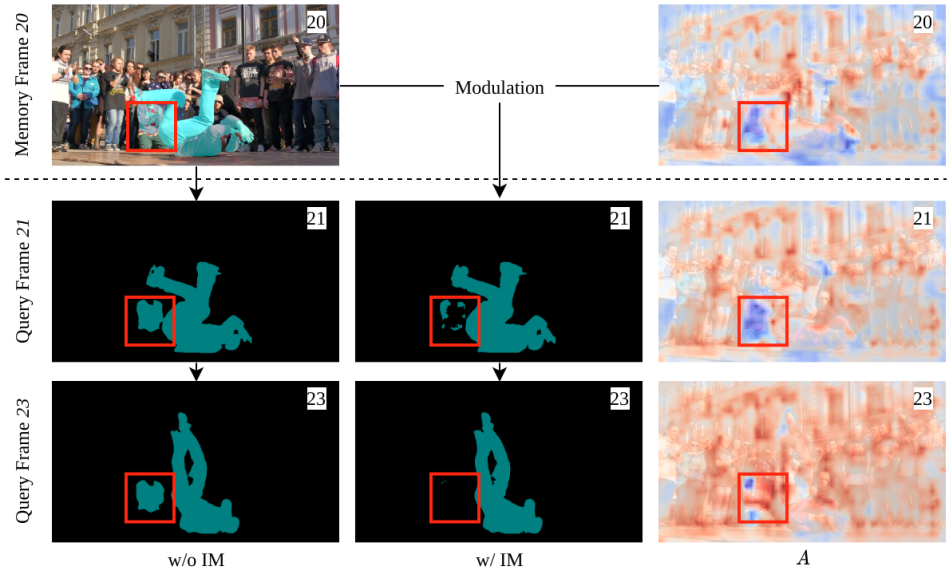

Figure 4. Segmentation results comparing whether to use IM module. The red bounding box in the memory frame highlights the wrongly segmented area. In the estimated A, lower values are represented in blue, while higher values are represented in red.

Positional Modulator: The PM module is devised to incorporate spatiotemporal positional information into the segmentation process. As depicted in Table 3, we initially explore the effectiveness of integrating spatiotemporal information within the VOS task. This is demonstrated by comparing models #1, #2 and #5. Among the models that incorporate positional information (i.e., #2 and #5), model #5, which integrates both spatial and temporal information, outperforms model #2, which solely utilizes spatial positional information. Additionally, all models incorporating positional information (#2 and #5) achieve higher accuracy than models lacking positional information (#1).

Subsequently, we delve into several design choices for the PM module. In our MMN, special consideration is given to the design of temporal coordinates and the corresponding normalization function. For comparison, we train model #3, where temporal coordinates are sequentially numbered, similar to the approach in VisTR [57]. Model #4 replaces the normalization function of our MMN with the commonly used Norm_M, as defined in Equation (6). As evidenced by Table 3, replacing either the temporal coordinates numbering scheme or the normalization function results in a degradation of segmentation accuracy. We conjecture that the combination of the coordinate numbering scheme and normalization using Equation (7) enables better utilization of information contained in two critical frames, namely, the first frame and the last frame.

Table 3. The results of ablation studies. We conduct an ablation study on the proposed Position Modulator (PM). 'Sequential' means temporal coordinates are numbered sequentially, while 'Customized' represents that temporal coordinates are numbered as proposed in the Position Modulator. Bold indicates the **best** performance.

	IM	PE	N	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
#1	✗	✗	✗	82.0	79.0	85.1
#2	✗	2D	Norm_M	83.4	80.3	86.5
#3	✗	3D-Sequential	Norm_M	83.5	80.4	86.6
#4	✗	3D-Customized	Norm_M	80.6	77.5	83.7
#5	✗	3D-Customized	Norm_R	84.0	80.9	87.2
#6	✓	✗	✗	84.3	80.5	88.1
#7	✓	3D-Customized	Norm_R	**85.8**	**82.8**	**88.9**

4.4. Comparison with State of the Art

DAVIS: The quantitative results of DAVIS 2017 and 2016 [11,62] are shown in Tables 4 and 5, respectively. On DAVIS 2016, the proposed MMN achieves a $\mathcal{J}\&\mathcal{F}$ score of 91.2% on the DAVIS 2016 validation set, which is comparable to other semi-supervised VOS solutions. On DAVIS 2017, which is multi-object-annotated, our MMN achieves a $\mathcal{J}\&\mathcal{F}$ score of 85.8%, which is comparable to other competitive state-of-the-art methods. From the table, it can be observed that the proposed MMN outperforms the baseline STM [3] by 3.6 in terms of region similarity score $\mathcal{J}$, 4.5 in terms of contour accuracy score $\mathcal{F}$ and 4.0 points in terms of average score $\mathcal{J}\&\mathcal{F}$. This is attributed to the MMN's adoption of the memory modulation mechanism, resulting in improvements in both pixel-level accuracy and boundary precision. Since the contour accuracy score $\mathcal{F}$ emphasizes boundary precision, the MMN demonstrates more significant improvements in this aspect. We also present qualitative results in Figure 5. Due to the modulated memory, the MMN manages to segment the challenging tail area of the motorcycle, while the baseline method STM [3] accumulates the segmentation error throughout the video.

Table 4. Quantitative evaluation on the DAVIS 2017 validation set. Bold and underline indicate the **best** and the second-best performance, respectively.

Methods	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
OnAVOS [38]	67.9	64.5	71.3
OSMN [28]	54.8	52.5	57.1
OSVOS [29]	59.2	56.6	61.8
RGMP [41]	63.2	64.8	68.6
FAVOS [68]	58.2	54.6	61.8
CINN [34]	70.7	67.2	74.2
VideoMatch [37]	62.4	56.5	68.2
PReMVOS [39]	77.8	73.9	81.7
A-GAME [32]	70.0	67.2	72.7
FEELVOS [35]	71.6	69.1	74.0
STM [3]	81.8	79.2	84.3
KMN [5]	82.8	80.0	85.6
EGMN [6]	82.9	80.0	85.9
CFBI [33]	81.9	79.1	84.6
AFB-URR [7]	74.6	73.0	76.1
RMNet [10]	83.5	81.0	86.0
SST [69]	82.5	79.9	85.1
LCM [9]	83.5	80.5	86.5
STCN [27]	85.4	82.2	88.6
BMVOS [31]	72.7	70.7	74.7
RDE-VOS [40]	84.2	80.8	87.5
DEVA [70]	**86.8**	**83.6**	**90.0**
Ours	<u>85.8</u>	<u>82.8</u>	<u>88.9</u>

Table 5. Quantitative evaluation on the DAVIS 2016 validation set. Bold and underline indicate the **best** and the second-best performance, respectively.

Methods	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	**FPS**
OnAVOS	85.5	86.1	84.9	0.1
OSVOS	80.2	79.8	80.6	0.1
MaskRNN	80.8	80.7	80.9	-
CINN	84.2	83.4	85.0	<0.1
LSE	81.6	82.9	80.3	-
VideoMatch	81.9	81.0	80.8	3.1
PReMVOS	86.8	84.9	88.6	<0.1
A-GAME	82.1	82.2	82.0	14.3
FEELVOS	82.2	81.7	88.1	2.2
STM	89.3	88.7	89.9	6.3
KMN	90.5	89.5	91.5	8.3
CFBI	89.4	88.3	90.5	5.6
RMNet	88.8	88.9	88.7	11.1
RDE-VOS	91.1	89.7	92.5	35.0
Ours	**91.4**	**90.6**	**92.2**	6.0

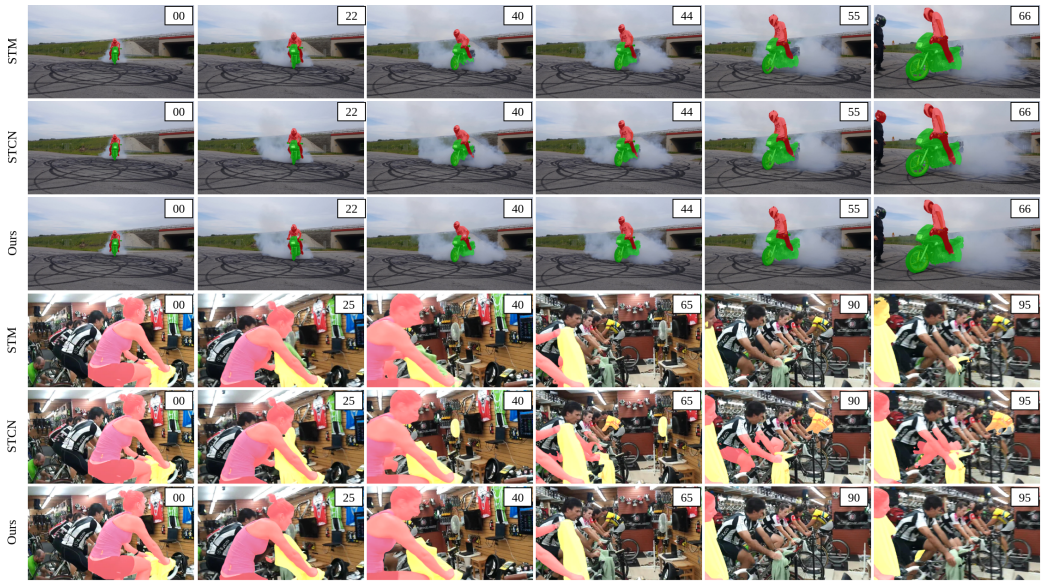

Figure 5. Visual comparison with the state-of-the-art methods.

YouTube-VOS: Due to the large amount of videos, the YouTube-VOS benchmark poses a huge challenge to VOS solutions. The quantitative results of the YouTube-VOS 2018 validation set are presented in Table 6, where the proposed MMN performs favorably against the state-of-the-art methods. From the table, it can be observed that the proposed MMN outperforms the baseline method STM [3] in terms of $\mathcal{J}$ score by 1.5 and $\mathcal{F}$ score by 2.3 for the seen category. For the unseen category, the MMN achieves an improvement of 4.8 in $\mathcal{J}$ score and 5.3 in $\mathcal{F}$ score. It benefits from the Position Modulator in the MMN, which enhances the positional information, thereby improving its performance particularly when encountering unseen object categories during training.

A visual comparison of segmentation results on YouTube-VOS video *37dc952545* is also provided in Figure 5. STM [3] loses track when the target head reappears at frame *90*, while STCN [27] is disturbed by similar objects from frame *40*. In contrast, with the proposed PM module, our MMN succeeds in relocating the target head at the frame, while avoiding interference by the background.

Table 6. Quantitative evaluation on the YouTube-VOS 2018 validation set. Bold and underline indicate the **best** and the second-best performance, respectively.

Methods	$\mathcal{G}$	$\mathcal{J}_s$	$\mathcal{F}_s$	$\mathcal{J}_u$	$\mathcal{F}_u$
OnAVOS	55.2	60.1	62.7	46.6	51.4
OSMN	51.2	60.0	60.1	40.6	44.0
OSVOS	58.8	59.8	60.5	54.2	60.7
RGMP	53.8	59.5	-	45.2	-
BoLTVOS	71.1	71.6	-	64.3	-
PReMVOS	66.9	71.4	75.9	56.5	63.7
A-GAME	66.1	67.8	-	60.8	-
STM	79.4	79.7	84.2	72.8	80.9
KMN	81.4	81.4	85.6	75.3	83.3
EGMN	80.2	80.7	85.1	74.0	80.9
CFBI	81.4	81.1	85.8	75.3	83.4
AFB-URR	79.6	78.8	74.1	83.1	82.6
RMNet	81.5	82.1	85.7	75.7	82.4
LCM	82.0	**82.2**	86.7	75.7	83.4
SST	81.7	81.2	-	76.0	-
STCN	83.0	81.9	86.5	**77.9**	85.7
BMVOS	73.9	73.5	68.5	77.4	76.0
Ours	**83.1**	81.2	86.5	77.6	**86.2**

MOSE: The MOSE [64] dataset presents a unique challenge due to its more intricate scenes and extended video lengths, which demand robust segmentation algorithms capable of maintaining accuracy over prolonged durations. While our MMN surpasses the baseline method STM [3] on MOSE, it lags behind DEVA's [70] results (Table 7). We attribute this discrepancy to the integration of an LSTM module designed explicitly for handling longer video sequences in DEVA. To address this, future improvements could involve augmenting the MMN architecture with a dedicated LSTM module tailored specifically for handling long video sequences.

Table 7. Quantitative evaluation on the MOSE validation set. Bold and underline indicate the **best** and the second-best performance, respectively.

Methods	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
STM [3]	48.6	44.2	52.9
RDE-VOS [40]	48.8	44.6	52.9
STCN [27]	50.8	46.6	55.0
DEVA [70]	**60.0**	**55.8**	**64.3**
Ours	52.1	46.9	57.2

5. Conclusions

This study presented the Modulated Memory Network (MMN) for the challenging task of one-shot video object segmentation. Our approach introduces two key components: the Importance Modulator (IM) and the Position Modulator (PM). The IM module plays a pivotal role in constructing an informative memory. It modulates memory embeddings while taking into account their contextual importance and segmentation confidence. By doing so, it enhances the quality of the memory and significantly reduces the propagation of errors during the segmentation process. Moreover, the PM module enhanced the way we incorporate spatiotemporal information into the segmentation process. Through a tailored encoding scheme, it empowers both memory and query embeddings with vital spatiotemporal information. This enhancement reinforces the accuracy of information retrieval, resulting in more precise and robust object segmentation. The experimental results prove that the proposed MMN performs favorably against the state-of-the-art methods on YouTube-VOS while achieving top-rank performance on the DAVIS validation split.

Author Contributions: Conceptualization, H.L. and W.Z.; methodology, H.L., Z.G. and W.Z.; software, H.L.; validation, H.L. and Z.G.; formal analysis, H.L., Z.G. and W.Z.; investigation, H.L.; resources, H.L.; data curation, H.L.; writing—original draft preparation, H.L.; writing—review and editing, Z.G. and W.Z.; visualization, H.L. and Z.G.; supervision, W.Z.; project administration, W.Z.; funding acquisition, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key Research and Development Program of China under grant no. 2021ZD0112100.

Data Availability Statement: COCO [65] is available at https://cocodataset.org/, YouTube-VOS [63] is available at https://youtube-vos.org/, and DAVIS [11] is available at https://davischallenge.org/ (accessed on 9 September 2018).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Weston, J.; Chopra, S.; Bordes, A. Memory networks. *arXiv* **2014**, arXiv:1410.3916.
2. Graves, A.; Wayne, G.; Danihelka, I. Neural turing machines. *arXiv* **2014**, arXiv:1410.5401.
3. Oh, S.W.; Lee, J.Y.; Xu, N.; Kim, S.J. Video object segmentation using space-time memory networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9226–9235.
4. Li, Y.; Shen, Z.; Shan, Y. Fast Video Object Segmentation using the Global Context Module. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 735–750.
5. Seong, H.; Hyun, J.; Kim, E. Kernelized Memory Network for Video Object Segmentation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 629–645.
6. Lu, X.; Wang, W.; Danelljan, M.; Zhou, T.; Shen, J.; Van Gool, L. Video Object Segmentation with Episodic Graph Memory Networks. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; Volume 12348, pp. 661–679.
7. Liang, Y.; Li, X.; Jafari, N.; Chen, J. Video Object Segmentation with Adaptive Feature Bank and Uncertain-Region Refinement. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–12 December 2020; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 3430–3441.
8. Wang, H.; Jiang, X.; Ren, H.; Hu, Y.; Bai, S. SwiftNet: Real-time Video Object Segmentation. *arXiv* **2021**, arXiv:2102.04604.
9. Hu, L.; Zhang, P.; Zhang, B.; Pan, P.; Xu, Y.; Jin, R. Learning Position and Target Consistency for Memory-based Video Object Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 4144–4154.
10. Xie, H.; Yao, H.; Zhou, S.; Zhang, S.; Sun, W. Efficient Regional Memory Network for Video Object Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021.
11. Pont-Tuset, J.; Perazzi, F.; Caelles, S.; Arbelaez, P.; Sorkine-Hornung, A.; Gool, L.V. The 2017 DAVIS Challenge on Video Object Segmentation. *arXiv* **2017**, arXiv:1704.00675.
12. Wang, W.; Shen, J.; Shao, L. Video salient object detection via fully convolutional networks. *IEEE Trans. Image Process.* **2017**, *27*, 38–49. [CrossRef] [PubMed]
13. Yang, C.; Lamdouar, H.; Lu, E.; Zisserman, A.; Xie, W. Self-supervised video object segmentation by motion grouping. In Proceedings of the the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 7177–7188.
14. Zhang, K.; Zhao, Z.; Liu, D.; Liu, Q.; Liu, B. Deep transport network for unsupervised video object segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 8781–8790.
15. Fragkiadaki, K.; Arbelaez, P.; Felsen, P.; Malik, J. Learning to segment moving objects in videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4083–4090.
16. Lu, X.; Wang, W.; Ma, C.; Shen, J.; Shao, L.; Porikli, F. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3623–3632.
17. Lee, M.; Cho, S.; Lee, S.; Park, C.; Lee, S. Unsupervised video object segmentation via prototype memory network. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–7 January 2023; pp. 5924–5934.
18. Cho, S.; Lee, M.; Lee, S.; Park, C.; Kim, D.; Lee, S. Treating motion as option to reduce motion dependency in unsupervised video object segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–7 January 2023; pp. 5140–5149.
19. Ponimatkin, G.; Samet, N.; Xiao, Y.; Du, Y.; Marlet, R.; Lepetit, V. A simple and powerful global optimization for unsupervised video object segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–7 January 2023; pp. 5892–5903.
20. Garg, S.; Goel, V. Mask selection and propagation for unsupervised video object segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–7 January 2021; pp. 1680–1690.

21. Ren, S.; Liu, W.; Liu, Y.; Chen, H.; Han, G.; He, S. Reciprocal transformations for unsupervised video object segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15455–15464.
22. Lu, X.; Wang, W.; Shen, J.; Crandall, D.; Luo, J. Zero-shot video object segmentation with co-attention siamese networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 2228–2242. [CrossRef] [PubMed]
23. Wang, W.; Lu, X.; Shen, J.; Crandall, D.J.; Shao, L. Zero-shot video object segmentation via attentive graph neural networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9236–9245.
24. Yang, S.; Zhang, L.; Qi, J.; Lu, H.; Wang, S.; Zhang, X. Learning motion-appearance co-attention for zero-shot video object segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1564–1573.
25. Zhen, M.; Li, S.; Zhou, L.; Shang, J.; Feng, H.; Fang, T.; Quan, L. Learning discriminative feature with crf for unsupervised video object segmentation. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXVII 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 445–462.
26. Gu, Y.; Wang, L.; Wang, Z.; Liu, Y.; Cheng, M.M.; Lu, S.P. Pyramid constrained self-attention network for fast video salient object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 10869–10876.
27. Cheng, H.K.; Tai, Y.W.; Tang, C.K. Rethinking Space-Time Networks with Improved Memory Coverage for Efficient Video Object Segmentation. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 11781–11794.
28. Yang, L.; Wang, Y.; Xiong, X.; Yang, J.; Katsaggelos, A.K. Efficient Video Object Segmentation via Network Modulation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
29. Caelles, S.; Maninis, K.K.; Pont-Tuset, J.; Leal-Taixe, L.; Cremers, D.; Van Gool, L. One-Shot Video Object Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
30. Maninis, K.K.; Caelles, S.; Chen, Y.; Pont-Tuset, J.; Leal-Taixé, L.; Cremers, D.; Van Gool, L. Video Object Segmentation without Temporal Information. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1515–1530. [CrossRef] [PubMed]
31. Cho, S.; Lee, H.; Kim, M.; Jang, S.; Lee, S. Pixel-level bijective matching for video object segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 129–138.
32. Johnander, J.; Danelljan, M.; Brissman, E.; Khan, F.S.; Felsberg, M. A generative appearance model for end-to-end video object segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 8953–8962.
33. Yang, Z.; Wei, Y.; Yang, Y. Collaborative video object segmentation by foreground-background integration. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020.
34. Bao, L.; Wu, B.; Liu, W. CNN in MRF: Video Object Segmentation via Inference in a CNN-Based Higher-Order Spatio-Temporal MRF. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5977–5986. [CrossRef]
35. Voigtlaender, P.; Chai, Y.; Schroff, F.; Adam, H.; Leibe, B.; Chen, L.C. FEELVOS: Fast End-To-End Embedding Learning for Video Object Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
36. Khoreva, A.; Benenson, R.; Ilg, E.; Brox, T.; Schiele, B. Lucid Data Dreaming for Object Tracking. In Proceedings of the 2017 DAVIS Challenge on Video Object Segmentation—CVPR Workshops, Honolulu, HI, USA, 21–26 July 2017.
37. Hu, Y.T.; Huang, J.B.; Schwing, A.G. Videomatch: Matching based video object segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 54–70.
38. Voigtlaender, P.; Leibe, B. Online adaptation of convolutional neural networks for the 2017 davis challenge on video object segmentation. In Proceedings of the 2017 DAVIS Challenge on Video Object Segmentation—CVPR Workshops, Honolulu, HI, USA, 21–26 July 2017; Volume 5.
39. Luiten, J.; Voigtlaender, P.; Leibe, B. PReMVOS: Proposal-generation, Refinement and Merging for Video Object Segmentation. In Proceedings of the Asian Conference on Computer Vision (ACCV), Perth, Australia, 2–6 December 2018.
40. Li, M.; Hu, L.; Xiong, Z.; Zhang, B.; Pan, P.; Liu, D. Recurrent dynamic embedding for video object segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 1332–1341.
41. Oh, S.W.; Lee, J.Y.; Sunkavalli, K.; Kim, S.J. Fast Video Object Segmentation by Reference-Guided Mask Propagation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
42. Xu, N.; Price, B.; Cohen, S.; Yang, J.; Huang, T.S. Deep interactive object selection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 373–381.
43. Cheng, H.K.; Tai, Y.W.; Tang, C.K. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 5559–5568.
44. Heo, Y.; Koh, Y.J.; Kim, C.S. Guided interactive video object segmentation using reliability-based attention maps. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7322–7330.

45. Oh, S.W.; Lee, J.Y.; Xu, N.; Kim, S.J. Fast user-guided video object segmentation by interaction-and-propagation networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5247–5256.
46. Heo, Y.; Jun Koh, Y.; Kim, C.S. Interactive video object segmentation using global and local transfer modules. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XVII 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 297–313.
47. Miao, J.; Wei, Y.; Yang, Y. Memory aggregation networks for efficient interactive video object segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10366–10375.
48. Wang, W.; Zhou, T.; Porikli, F.; Crandall, D.; Van Gool, L. A Survey on Deep Learning Technique for Video Segmentation. *arXiv* **2021**, arXiv:2107.01153.
49. Li, Z.; Jiang, J.; Chen, X.; Laganière, R.; Li, Q.; Liu, M.; Qi, H.; Wang, Y.; Zhang, M. Dense-scale dynamic network with filter-varying atrous convolution for semantic segmentation. *Appl. Intell.* **2023**, *53*, 26810–26826. [CrossRef]
50. Ou, X.; Wang, H.; Zhang, G.; Li, W.; Yu, S. Semantic segmentation based on double pyramid network with improved global attention mechanism. *Appl. Intell.* **2023**, *53*, 18898–18909. [CrossRef]
51. Yu, Z.; Lee, F.; Chen, Q. HCT-net: Hybrid CNN-transformer model based on a neural architecture search network for medical image segmentation. *Appl. Intell.* **2023**, *53*, 19990–20006. [CrossRef]
52. Hu, Y.T.; Huang, J.B.; Schwing, A. MaskRNN: Instance Level Video Object Segmentation. In *Proceedings of the Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
53. Perazzi, F.; Khoreva, A.; Benenson, R.; Schiele, B.; Sorkine-Hornung, A. Learning Video Object Segmentation from Static Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3491–3500. [CrossRef]
54. Yang, Y.; Gu, X. Learning rich feature representation and aggregation for accurate visual tracking. *Appl. Intell.* **2023**, *53*, 28114–28132. [CrossRef]
55. Shin Yoon, J.; Rameau, F.; Kim, J.; Lee, S.; Shin, S.; So Kweon, I. Pixel-level matching for video object segmentation using convolutional neural networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2167–2176.
56. Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; Lillicrap, T. Meta-learning with memory-augmented neural networks. In Proceedings of the International Conference on Machine Learning (ICML), PMLR, New York, NY, USA, 19–24 June 2016; pp. 1842–1850.
57. Wang, Y.; Xu, Z.; Wang, X.; Shen, C.; Cheng, B.; Shen, H.; Xia, H. End-to-End Video Instance Segmentation with Transformers. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021.
58. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.
59. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vegas, NV, USA, 27–30 June 2016.
60. Glorot, X.; Bordes, A.; Bengio, Y. Deep Sparse Rectifier Neural Networks. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
61. Milletari, F.; Navab, N.; Ahmadi, S.A. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 565–571. [CrossRef]
62. Perazzi, F.; Pont-Tuset, J.; McWilliams, B.; Van Gool, L.; Gross, M.; Sorkine-Hornung, A. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 724–732. [CrossRef]
63. Xu, N.; Yang, L.; Fan, Y.; Yang, J.; Yue, D.; Liang, Y.; Price, B.; Cohen, S.; Huang, T. Youtube-vos: Sequence-to-sequence video object segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 585–601.
64. Ding, H.; Liu, C.; He, S.; Jiang, X.; Torr, P.H.; Bai, S. MOSE: A New Dataset for Video Object Segmentation in Complex Scenes. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023.
65. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
66. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 248–255.
67. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.

68. Cheng, J.; Tsai, Y.H.; Hung, W.C.; Wang, S.; Yang, M.H. Fast and Accurate Online Video Object Segmentation via Tracking Parts. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
69. Duke, B.; Ahmed, A.; Wolf, C.; Aarabi, P.; Taylor, G.W. SSTVOS: Sparse Spatiotemporal Transformers for Video Object Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 5912–5921.
70. Cheng, H.K.; Oh, S.W.; Price, B.; Schwing, A.; Lee, J.Y. Tracking Anything with Decoupled Video Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023.