Unsupervised video object segmentation: an affinity and edge learning approach

Sundaram Muthu1 · Ruwan Tennakoon2 · Reza Hoseinnezhad1 · Alireza Bab-Hadiashar1

Received: 28 September 2021 / Accepted: 14 July 2022 / Published online: 10 August 2022
© The Author(s) 2022

Abstract
This paper presents a new approach, called TMNet, to solve unsupervised video object segmentation (UVOS) problem. The UVOS is still a challenging problem as prior methods suffer from issues like generalization errors in unseen test videos, over reliance on optic flow, and capturing fine details at object boundaries. These issues make the UVOS an ill-defined problem, particularly in presence of multiple objects. Our focus is to constrain the problem and improve the segmentation results by fusion of multiple available cues such as appearance and motion, as well as image and flow edges. To constrain the problem, instead of predicting segmentation directly, we predict affinities between neighbouring pixels for being part of the same object and cluster those to obtain category agnostic segmentation. To further improve the segmentation, we fuse multiple-sources of information through a novel Temporal Motion Attention (TMA) module that uses neural attention to learn powerful spatio-temporal features. In addition, we also design an edge refinement module (using image and optic flow edges) to refine and improve the accuracy of object segmentation boundaries. The overall framework is capable of segmenting and finding accurate objects’ boundaries without any heuristic post processing. This enables the method to be used for unseen videos. Experimental results on challenging DAVIS16 and multi object DAVIS17 datasets show that our proposed TMNet performs favorably compared to the state-of-the-art methods without post processing.

Keywords Video object segmentation · Neural attention · Edge refinement · Affinity learning · Multi-cue segmentation · Correlation clustering

1 Introduction
Video object segmentation (VOS) is a common task in many video analysis and scene understanding applications such as video editing [1], autonomous driving [2], robotics, surveillance and tracking [3]. It involves pixel level segmentation of independent moving objects across frames in a given video sequence.

Traditionally, VOS (also called motion segmentation) is solved via fitting geometric models to matched key-points in adjacent frames [4]. More recently, deep learning-based methods have become prominent. Existing deep learning-based VOS methods can be classified as semi-supervised, interactive and unsupervised methods based on the amount of human involvement in the segmentation process. The semi-supervised methods (SVOS) require annotations of objects of interest in the first frame. The interactive methods require user interactions like scribbles, to guide and correct the segmentation. Recently introduced unsupervised VOS methods (UVOS) are expected to identify all moving objects in the scene, with no prior information about the number of objects or manual annotation of the first frame.

Unsupervised video object segmentation is challenging since the segments or the number of segments are not unknown at start. Other challenges include dynamically varying number of objects, camouflaged object motions, occlusions, articulated non-rigid object motions, and background motion.

Prior works follow three main strategies to solve the UVOS problem: (1) using motion and appearance features together; (2) using semantic segmentation of objects in a video frame, followed by tracking of the detected objects using temporal information in subsequent video frames; and
(3) using neural attention to leverage temporal information from optic flow to focus on obtaining better features that represent moving objects.

Methods using motion and appearance features together [5–8] generally rely on two stream architectures. i.e., processing image (appearance features), and optic flow (motion features) independently. The performance of the two stream architectures are strongly dependent on accuracy of the data correspondences provided by optic flow which is often poor around object boundaries [9], non-textured regions, and fast-moving objects [10]. In such cases, motion features would not be reliable to complement the appearance features for accurate segmentation. This leads to an over-reliance of these methods towards the appearance features of the objects.

Object detection and tracking methods [11–14] use state-of-the-art object detection methods like Mask-RCNN [15] to detect foreground objects and track all detected objects using tracking algorithms. Tracking has its own challenges: it often suffers from drift, and it relies heavily on object re-identification to track missed or re-appearing objects [16]. These methods usually suffer from generalization errors, when applied to larger test videos containing objects that do not appear in the training data.

To overcome the above challenges, in a solution called Motion-Attentive Transition Network (MATNet) [17], a neural attention mechanism was introduced that is similar to how humans perform motion segmentation. The motion-attentive module uses optic flow to focus on moving objects and to obtain better features. This method demonstrated improved performance in tackling the above mentioned dilemmas, but it has its own limits and room for further improvement. For instance MATNet’s primary focus on foreground/background segmentation limits its applicability in scenarios where multiple moving objects need to be segmented. In addition, the optic flow information employed by MATNet is drawn from only two consecutive frames. This introduces limited temporal information. Finally, fine details of objects are not entirely captured by optic flow, leading to poor object boundaries.

To solve the UVOS problem, we propose a novel approach, called TMNet, that combines temporal, edge and affinity information. The integration of various cues is performed in the sequence shown in Fig. 1 and produces segmentation affinities that can be clustered to segments that are significantly more accurate than those returned by state-of-art.

The design shown in Fig. 1 is developed to ensure that several desirable outcomes are achieved with significantly enhanced accuracy compared to the state-of-art. A major enabling factor here is the inclusion of several consecutive optic flows as the inputs to TMNet. It is common sense that with more information employed, better accuracy is expected. However, it is the design and implementation of the overall system and its details to a surgical precision that make the resulting accuracy enhancements feasible. More precisely, it is the choice of the operations (blocks) and the sequence in which they operate, as well as the procedures executed within each block that all together, make the difference. The resulting implementation that will be presented in next sections in detail, deliver the following advantages.

Firstly, multiple objects can now be segmented in video as affinities are employed instead of segmentation masks. This stems from assessing the relationship between neighboring pixels by assigning them the probability of

![Fig. 1 Overall schematics of our proposed UVOS solution called TMNet. Temporal attention encoder block captures spatio-temporal information from the current video frame (I) and consecutive optic flows (V_{t-1}, \ldots, V_{t-n}). Affinity learning decoder block predicts neighbour affinities using the features learnt by the encoder. Edge enhancement block refines the object segmentation boundaries by aligning and merging the image edge, temporal flow edge and affinity information using non-linear aggregation functions. Correlation clustering block converts the edge-enhanced affinity graph into the required segmentation output.](image_url)
belonging to either the same object (when pixels lie inside an object) or a different one (when pixels straddle object boundaries).

Secondly, the proposed design is capable of detecting and segmenting highly complex object motions with continuing and fluid changes in their appearance through time. This is enabled by by encoding temporal attention from image and consecutive optic flow information. The idea is inspired by the observation that humans are attracted first to anything that moves before learning to map objects to semantic object classes [18].

Thirdly, TMNet is less prone to inaccuracies at moving object boundaries. This is achieved through the edge enhancement block. The edge enhancement block aligns the output of image edge network, temporal flow edge network and the affinity learning decoder network through a series of simple conv blocks. The conv blocks act like a non-linear aggregate function that merges the image edge, flow edge and affinity information in order to refine the edges of the segmentation.

Figure 2 presents instances of motion segmentation executed on DAVIS17 dataset to demonstrate how the above three areas of improvement are achieved.

The main contributions of this paper include:

- A new Temporal Motion Attention (TMA) module within the Temporal Attention Encoder block to obtain powerful spatio-temporal features for segmenting moving objects.
- Predicting neighbourhood affinities instead of directly predicting the segmentation to obtain category agnostic segmentation of multiple moving objects in unseen videos.
- An edge refining network, within the Edge Enhancement block, for combining the image and flow edges, and affinity information to refine the object segmentation boundaries.

This paper is organized as follows: Sect. 2 introduces the related works. Section 3 presents the network architecture of TMNet in more detail, including its implementation and the loss function used. Ablation studies on DAVIS16 dataset are presented in Sect. 4. They demonstrate the performance improvements achieved. Section 5 concludes the paper and discusses future work.

2 Related works

2.1 Definition and categorization

According to Gestalt’s “common fate” principle [19], VOS is the grouping of pixels with the same motion. Torr [20] defines VOS as segmenting all objects that move relative to the background. This is ambiguous since in many applications, motion grouping may not always be equivalent object grouping. For example, this may occur in applications where there are intermittent object motions (objects are static for few frames in the sequence), articulated objects (only part of the object moves), or similarly moving objects (different objects with same motion).

Bideau et al. [21] analyzed various applications and identified challenging cases similar to above, then proposed that VOS is defined as grouping moving objects in a way that (i) an entire object is segmented even if only part of it moves, (ii) a temporarily static object is segmented due to past recent movements; and (iii) similarly moving objects are segmented separately unless they are connected in 3D.

Depending on the amount of human involvement, a VOS solution can be classified as semi-supervised, interactive or unsupervised. Unlike semi-supervised methods that require first frame ground-truth of objects to be segmented, unsupervised or zero shot VOS methods should identify all moving object(s) without any prior information. Note that the work...
reported in this paper addresses the challenging UVOS problem that requires the model to have generalizing capability.

2.2 UVOS based on multiple cues

It is well-known that UVOS cannot perform accurately if only a single source of visual information is extracted from moving images and applied. For instance, appearance-based methods would fail to segment texture-less objects and would segment static objects in the background. On the other hand, the solutions that focus on motion and merely use optic flow information may over-segment non-rigid objects, and segmenting occluded or camouflaged objects and those with degenerate motions.

Inspired by the above observations, a number of UVOS solutions have been recently developed based on combining the appearance and motion (optic flow) information [22–24]. Due to inaccuracy of optic flow at object boundaries, Koh et al. [25] and Papazoglou and Ferrari [26] proposed to employ edge cues to obtain better performance at object boundaries. Differently, reinforcement learning has also been used to solve the UVOS task [27].

Recent advances in deep learning for object recognition [15, 28] have enabled the use of temporal information to track object proposals and generate consistent segmentation for the entire video [11, 13, 29, 30]. Zhao et al. [31] have suggested to perform UVOS for multiple objects by detecting and tracking objects using human-centric re-identification. In another solution called unsupervised offline video object segmentation and tracking (UnOViST) [11], tracklets are generated for object proposal masks and long-term consistent tracklets are merged to perform segmentation. A propose-reduce paradigm [30] extends UnOViST [11] by improving the merging step using only key-frames selected by some heuristic rules. Object-based detection methods use Siamese re-identification networks for association. This can lead to failure for detecting fast moving objects, occlusions, and non-rigid motions [16].

In contrast to above methods, our approach effectively combines all the cues discussed above to produce category agnostic multi-object UVOS. Most methods perform only binary foreground/background segmentation, thereby limiting their applicability to scenes with only one moving object. In contrast, we perform multi-object UVOS by predicting affinities instead of predicting the segmentation directly. These are explained in Sect. 3.

2.3 Attention in neural networks

Inspired by human perception, detection of visual attention has been recently implemented using deep neural networks to improve the performance of machine vision solutions for various applications such as attention guided object segmentation [32], dynamic visual attention prediction [17], depth estimation [33], action recognition [34], and visual question answering [35]. Attention helps the network form effective feature representations from the data. The effectiveness is achieved by focusing only on the relevant informative regions of interest (avoiding unnecessary information). Neural attention has also been used for improving the performance of the UVOS solutions in [17, 36–39].

In order to avoid the use of computationally expensive optic flow for UVOS, AGNN [36] and AGS [38] first use an attention mechanism to capture the higher order relationships in a message passing graph neural network framework. Differently, COSNet [37] solved UVOS using co-attention between frames in a video sequence by a Siamese neural network to learn the global context. MATNet [17] introduced a motion attentive two stream interleaved encoder to learn powerful spatio-temporal features for UVOS by an attention mechanism using optic flow to focus on only the moving objects. FEM-Net [39] extended MATNet [17] by additionally using the optic flow edge information in a Flow edge connect module that helps in segmenting the foreground salient objects, and their boundaries accurately. Our method also extends the motion-based attention mechanism proposed by MATNet [17] to learn powerful spatio-temporal features by additionally incorporating temporal context from several frames in the video into the attention mechanism. This helps to resolve ambiguities in optic flow information of complex dynamic scenes in two consecutive frames.

Differently from the previous methods, our method also predicts affinities instead of predicting the output segmentation directly. This enables our method to be used for category agnostic multiple object segmentation.

3 Proposed solution: TMNet

3.1 Problem statement

Consider a video sequence represented by its frames

\[ I = (I_1, I_2, \ldots, I_T) \in \mathbb{R}^{W \times H \times 3 \times T} \]

where each video frame, \( I_t \), denotes an RGB image with width \( W \) and height \( H \) in pixels. We denote the forward optic flow computed from those frames by

\[ V = (V_1, V_2, \ldots, V_{T-1}) \in \mathbb{R}^{W \times H \times 2 \times (T-1)}. \]

The objective of VOS is to generate the multi-object segmentation

\[ S = \{S_1, S_2, \ldots, S_T\} \]
where $S_t$ denotes the segmentation at frame $t$ and is itself a set of binary object segmentation masks,
\[
S_t = \left\{ M_t^{(1)}, M_t^{(2)}, \ldots, M_t^{(K_t)} \right\}
\]
and each mask is a $W \times H$ matrix of object labels in $\mathbb{L}_t$. Note that both the number of moving objects, $K_t$, and the space of possible object labels (discovered so far), $\mathbb{L}_t$, depend on the frame number $t$, as they can dynamically vary in the video sequence due to new objects entering or outgoing objects leaving the scene.

Notations: The notations used throughout this paper are as follows. $\mathbb{R}$ denotes the Space of real numbers. $W$ and $H$ represent the width and height of the image respectively. $I_t$ denotes the colour image at time $t$. $V_t$ denotes the optic flow at time $t$. $\delta$ is the hyperparameter that denotes the number of consecutive optic flows used. $F_{i,t}$ and $\hat{F}_{i,t}$ denotes appearance and temporal features extracted at the $i$-th residual stage at time $t$ respectively. $F_t$ denotes the temporal motion attentive feature obtained from the encoder. $A_t$ denotes the segmentation affinity matrix obtained from the decoder. $I_t$ and $V_t$ represent the image edge and flow edge maps respectively. $\hat{A}$ denotes the refined affinity matrix after edge enhancement. Finally, $S_t$ denotes the segmentation output at time $t$.

Our proposed solution to this problem, TMNet, is an end-to-end deep neural network with its overall architecture presented in Fig. 1. At its core, TMNet predict affinities by learning powerful spatio-temporal features (extracted from video and consecutive optic flows) through neural attention and enhancing them around the edges. The internals of the network are described in the following subsections.

### 3.2 Temporal attention encoder

This encoder uses a motion attentive two stream interleaved architecture to learn robust feature representations for moving object(s). We can achieve a degree of robustness due to the use of our novel temporal motion attention (TMA block) described in Sect. 3.3 that uses a neural attention mechanism to focus on only the moving object(s) (Fig. 3).

For the appearance stream, we use the image $I_t \in \mathbb{R}^{W \times H \times 3}$ and for the motion stream, we use the optic flows calculated between the current frame $t$ and $\delta$ previous frames $V_t = (V_{t-1}, \ldots, V_{t-\delta}) \in \mathbb{R}^{W \times H \times 2 \times \delta}$. The optic flows are converted to RGB domain before passing through the convolutional layers ($\hat{V}_t \in \mathbb{R}^{W \times H \times 3 \times \delta}$).

Generally, as an encoder, this block would process the multi-cue information from the inputs and produce a robust feature representation $F_i \in \mathbb{R}^{W \times H \times C}$:
\[
F_i = \text{ENCODER}(I_i, \hat{V}_i).
\]
However, to enable the integration of additional temporal information from the previous $\delta$ frames, several innovations are in place.

Firstly, we use the initial 5 convolutional blocks of the standard ResNet backbone to extract appearance feature $F_{i,t}$ from the image $I_t$, and temporal features $(F_{m_1,t}, \ldots, F_{m_\delta,t})$ from the optic flows $\hat{V}_t$ at various residual stages ($i = 2, \ldots, 5$) with different spatial resolutions (1/2th, 1/4th, 1/8th, 1/16th) of the original image size. Note that spatial resolution of the $i$-th stage is indeed $1/2^{i-1}$ of the original image size.

Next, we use the temporal features to update the appearance features at all intermediate stages, and obtain the enhanced appearance features $\tilde{F}_{i,t}$ as output features that would be employed for affinity learning and prediction. This enhancement is performed using a newly designed TMA block for each stage:
\[
\tilde{F}_{i,t} = \text{TMA} \left( F_{i,t}, (F_{m_1,t}, \ldots, F_{m_\delta,t}) \right).
\]

### 3.3 The TMA block

Figure 4 shows a schematic diagram demonstrating how appearance and temporal features are combined in the TMA block. Note that in the diagram, signals appear inside boxes and operations (such as convolutions, fully connected layers and multiplications) appear outside blocks and arrows. In addition, for the sake of simplicity, we have removed the frame time $t$ and stage index $i$ from the signal notation.

A temporal aggregation block within our TMA block to extend the Motion Attentive Transition block developed...
by MATNet [17] so that temporal information from previous $\delta$ frames can be accommodated. Firstly, there is a soft attention that weights each of feature maps ($F = F_a, \{F_m1, \ldots, F_m\delta\}$) at every pixel location. This is performed by a 1x1 conv function that learns the probability that a particular region of the feature map is important. The probability is then normalized using a softmax function to obtain the normalized importance weights $I \in \mathbb{R}^{W \times H}$. The feature maps in $F$ are all converted separately to obtain the spatial attentive features $Z$ by using a channel-wise Hadamard product:

$$Z^c_a = \text{SOFTMAX}(W_a(F_a)) \odot F^c_a. \quad (4)$$

Next, we find the correlation between the spatial attentive appearance features $Z_a$, and each of the spatial attentive motion features $Z_m$. This correlation $S_i$, or the non-linear affinity, is learnt to find the relationship between the two feature spaces. The affinity is high (both appearance and temporal motion features are similar) for regions of the image where moving object(s) are present:

$$S_i = Z^T_m (W_i) Z_a \quad S_i \in \mathbb{R}^{WH \times WH}$$

$$= Z^T_m (P, Q^T) Z_a \quad P, Q \in \mathbb{R}^{C \times \frac{C}{2}}$$

$$= (P^T Z_m) (Q^T Z_a) \quad i \in \{1, \ldots, \delta\}. \quad (5)$$

where $P$ and $Q$ are trainable weights learnt during training to compress the size of the model and avoid the overfitting problem. $Q$ matrix which is learnt to compress the appearance features, are shared in the calculation of all $S_i$.

In the third step, we normalize the affinity matrix $S_i$ along the row dimension to ensure that the sum of the contributions of all channels is 1. Finally we aggregate the normalized affinities $S'_i$ for all $\delta$ optic flows to obtain the temporal motion attention factor $S$.

$$S'_i = \text{SOFTMAX}(S_i)$$

$$S = \text{AGGREGATION}(S'_1, \ldots, S'_\delta). \quad (6)$$

The aggregation function can be implemented by simple maximum, minimum, average or median values.
Experiments in Sect. 4 indicate that the averaging performs better than other functions.

Finally, the enhanced appearance features \( \hat{F}_a \in \mathbb{R}^{W \times H \times C_a} \) are obtained from the temporal motion attention factor \( S \) by

\[
\hat{F}_a = F_a \times S. \quad (7)
\]

The enhanced appearance features \( \hat{F}_a \) and the motion features \( F_m \) are concatenated to form the combined features. The combined features of all four residual stages are scaled to the size of the image, and combined to obtain the final robust feature representation \( F \). Finally, the features \( F \) are fed to the affinity learning decoder stage.

### 3.4 Affinity learning decoder

The affinity learning decoder is designed to use the temporal motion attentive features \( F_t \in \mathbb{R}^{W \times H \times C_t} \), and the predicted affinities \( A_t \in \mathbb{R}^{W \times H \times 4} \).

\[
A_t = \text{DECODER}(F_t). \quad (8)
\]

The segmentation affinity is defined for every pixel \( u \in I_e \), and one of its neighbouring pixels \( v \in N(u) \). The affinity describes the probability that the selected pixel, and its neighbour belongs to the same motion. The value varies from 0 to 1. The affinity is 1 if the label of the pixel matches its neighbor (high affinity for edges between pixels within the same segmentation). The affinity is 0 if the labels do not match (low affinity for edges between pixels that belong to different segments). This representation is permutation invariant and has a fixed size, making it easy to use for training purposes. Here we face the trade-off between accuracy and memory/time. Increasing the neighborhood size will return more accurate results at the expense of the large memory footprint of segmentation affinity leading to more prediction time. Hence, we restrict our model to predict affinities for only the four immediate neighboring pixels.

The correlation module is used for learning the segmentation affinities from the temporal motion attentive features. The correlation module calculates the correlations between the spatio-temporal features \( F_t \) of the image \( I_e \), and warped features \( F_t^{\text{shifted}} \) of the same image. The warping is performed by shifting the features to its four neighbouring nodes (left, right, top and bottom nodes). For calculating the correlations, we utilize the simple cosine similarity function instead of learning from a cost volume constructed from the features (since constructing the full cost volume makes model large in-terms of memory). Figure 5 shows the internal design of the affinity learning decoder.

### 3.5 Edge enhancement network

The edge enhancement network is designed to refine the object boundaries by updating the predicted affinities \( A_t \) obtained from the decoder. In order to capture fine details and improve the accuracy of object segmentation boundaries, the edge enhancement network aligns and merges three sources of complementary information available: (1) Image edge \( I_e \in \mathbb{R}^{W \times H} \) of the image \( I_e \), (2) Flow edge \( V_e \in \mathbb{R}^{W \times H} \) of the optic flow \( V_{t-1} \), and (3) Predicted affinities \( A_t \in \mathbb{R}^{W \times H \times 4} \) obtained from the decoder by using a non-linear aggregation function \( \text{REFINE}(\cdot) \):

\[
\hat{A} = \text{REFINE}(A_t, I_e, V_e). \quad (9)
\]

Figure 6 shows the motivation for using both the image edge \( I_e \), and the flow edge \( OF_e \) information jointly in the edge refinement module of our TMNet model. The first row shows failure of using optic flow edges \( V_e \) to detect the swan (inaccurate boundaries, background noise due to moving water). As our proposed method uses additional image edge cues, it can overcome this issue and segment the swan correctly. The second row shows failure of using image edges \( I_e \) to segment the moving car (due to the object having no texture). Again, our proposed method has been able to detect the car correctly using the flow edge information. The above two cases highlight the fact that both image and flow edges act as complementary information aiding the segmentation together to refine the boundaries. The third row also shows the improvement due to the complementary information even though both flow and image edges are accurate.
The non-linear aggregation function $\text{REFINE}()$ is implemented by series of $\text{conv}$ blocks on concatenated features. Firstly, we concatenate the three sources of input information: (1) Image edge $I_e \in \mathbb{R}^{W \times H}$, (2) Flow edge $V_e \in \mathbb{R}^{W \times H}$, and (3) Predicted affinities $A_t \in \mathbb{R}^{W \times H \times 4}$ to form concatenated edge features $R_t$:

$$R_t = \text{CONCAT}(A_t, I_e, V_e) \in \mathbb{R}^{W \times H \times 6}$$

Secondly, the concatenated edge features $R_t$ form the input to the convolutional modules. The input $R_t$ has a spatial dimension of $W \times H$ and six feature channels. We use three $\text{conv}$ blocks with a size of 32, 32 and 4 as the output channel dimension. We use the stride of one to maintain the aligned spatial dimension of $W \times H$. The spatial dimensions of the input do not change as we do not use any pooling layers. The convolution blocks perform non-linear mapping and aggregation at a higher dimensional space to capture the fine details of the boundaries. Figure 7 depicts how the above mentioned operations take place inside the edge enhancement block. The output refined affinity $\hat{A}$ at the end of the block is clustered to obtain the required boundary aware segmentation.

### 3.6 Predicted affinity to video object segmentation

The refined edges are finally clustered to obtain the required segmentation. Our network is not limited the specific set of object classes present in the training class since we predict affinities. But in-order to obtain the required segmentation $S$ from the affinity $\hat{A}$ predicted by our TMNet model, we need to perform a clustering step. Unlike other clustering methods, correlation clustering finds the optimal number of
clusters automatically. So we apply correlation clustering on
a pixel grid graph that uses the predicted affinities.

Firstly, we create a pixel grid graph $G = (V, E, W)$ for the
image $I_t$, from predicted affinities $\hat{A}$ as follows:

- Nodes $V$: Set of $N = W \times H$ vertices for every pixel in
the image $I_t$. $W$ and $H$ denote the width and height of the
image $I_t$.
- Edges $E$: Set of edges $e_{uv} \in \mathbb{R}^{N \times 4}$ connect four
neighbouring nodes $v$ (left, right, top and bottom nodes) of
every node $u$ that form the pixel grid.
- Weights $W$: The affinities $\hat{A} \in \mathbb{R}^{N \times 4}$ predicted by our
model are used as weights $w_{uv}$ for every edge defined in
the graph. It is the cost associated with assigning the two
nodes $u$ and $v$ of the edge $e_{uv}$ to distinct components.

Next, the segmentation is performed by solving the optimiza-
tion problem defined on the pixel grid graph created using
the predicted affinities as edge weights. The correlation clus-
tering or the graph multicut optimization problem is solved
using the method described in [40]. The output is a unique
decomposition of the graph $G$, which assigns 0/1 labels to all
the edges. Edges labelled 1 straddle distinct clusters.

Finally, once edges straddling distinct clusters are iden-
tified, the clusters can be separated to obtain the output
segmentation $S_t$ as the set of binary object segmentation
masks $\{M_1, M_2, \ldots, M_k\} \in \mathbb{R}^{W \times H}$. Our method automatically
assigns the optimal number of independently moving
objects $k$ which can vary dynamically in the video sequence.

### 3.7 Loss function

Since our network predicts affinities instead of the segmenta-
tion directly, we use a loss function based on the predicted
affinities. For the training, we formulate our loss between the
predicted affinities and the ground-truth affinity (instead of
the usually used binary cross entropy loss between the
predicted and the ground-truth segmentation).

Firstly, in-order to define a loss term, we need to con-
vert the labelled ground-truth segmentation to ground-truth
affinities. The ground-truth affinity is 1 if the label of the
pixel in the image, and its neighbour are the same. Consider
an image $I_t \in \mathbb{R}^{W \times H \times 3}$, and its segmentation ground-truth
$L_t \in \mathbb{R}^{W \times H}$. The ground-truth affinity matrix $A \in \mathbb{R}^{W \times H \times W \times H}$ is
then defined for each pixel $u \in I_t$ and one if its neighbouring
pixels $v \in N(u)$, as follows:

$$A_{uv} = \begin{cases} 1, & \text{if } L(u) = L(v), \\ 0, & \text{otherwise,} \end{cases}$$  

where, $W$ & $H$ are the width and height of the image, $M$ is
the number of pixels in the neighbourhood of a pixel $u$.

For the loss function, we use the mean square error (MSE)
function between the ground-truth affinities $A$ defined pre-
viously, and the affinities $\hat{A}$ predicted by our TMNet model
given by:

$$L(A, \hat{A}) = \frac{1}{n} \sum_{u \in V} \sum_{v \in N(u)} (1 - A_{uv}) L_e(A_{uv} - \hat{A}_{uv}) + \frac{1}{n} \sum_{u \in V} \sum_{v \in N(u)} A_{uv} \times L_{ne}(A_{uv} - \hat{A}_{uv}, w).$$  

We split the loss into two parts to overcome the imbal-
ance problem in the ground-truth affinity as number of 0’s
(edges - labels of the compared pixels do not match each
other) << number of 1’s (non-edges - labels of the compared
pixels match each other) in $A$. The first term is the normal
mean square error loss for 0’s (edges). The second term is
the weighted mean square error loss for 1’s (non-edges). The
equations for those losses are as follows:

$$L_e(x) = x^2$$  

$$L_{ne}(x, w) = w \times x^2.$$  

The weights $w \in \mathbb{R}^{W \times H}$ are the edge weights generated
using the normalized gradient magnitude of the image. We
incorporate the use of geometry to control the importance
of non-edge pixels during training. This loss penalizes edges
in an image, that are non-edges in $A$ (penalize static object
boundaries that do not appear in the motion boundaries).

Normalization terms $\alpha_e$ and $\alpha_{ne}$ count the number of
the edges and weighted sum of the non-edges in $A$, respectively.

$$\alpha_e = \sum_{u \in V} \sum_{v \in N(u)} (1 - A_{uv}) \quad \text{and}$$

$$\alpha_{ne} = \sum_{u \in V} \sum_{v \in N(u)} A_{uv} \times W_u.$$  

### 3.8 Implementation details

Our TMNet model is end-to-end trainable to predict affini-
ties that are clustered to obtain the required segmentation.

Training: For pre-processing, the images are scaled to
$384 \times 512 \times 3$. We also augment training data to prevent
over-fitting. We use the open-source Flownet2 [41] for
optic flow estimation. We also use RCF [42] for obtaining
image and flow edge maps. For fair comparison, we adopt
the same method for generating optic flow and edge maps
in all of our experiments. We train the model only on the 30
training set video sequences of the DAVIS17 dataset [43]
without the use of any additional training data. The model
is trained from the scratch using the loss term and affini-
ty ground-truth as explained in Eq. (11) in a supervised
manner using random initial weights. We train the network
with a batch size of 2. We follow a learning rate of $10^{-4}$ for
pre-training, and $10^{-5}$ for fine-tuning as the training schedule using the ADAM optimizer. The number of previous frames for extracting the temporal attention $\delta$ is chosen to be 3. The number of neighbours $M$ for every pixel used to calculate the affinity is chosen to be the 4 immediate neighbours. We choose the values to be as low as possible since increasing both $\delta$ and $M$, results in increased accuracy in the output segmentation at the cost of increased model capacity and run-time.

We have used traditional neural networks with iterative training mechanisms for updating the weights of the proposed model. The use of non-iterative training mechanisms can optimize the time-consuming training process and lead to faster convergence [44]. For instance, Cao et al. [45] proposed a randomized neural network called Bidirectional stochastic configuration network (BSCN) that can perform effective training for regression problems even in absence of a GPU. It will be interesting to study the improvement in the efficiency of our TMNet using BSCN as we also deal with a similar affinity regression problem.

Testing: For testing, we apply our trained TMNet model to the unseen videos. We use the current image and the optic flow of the previous $\delta$ frames to produce the output segmentation. It is to be noted that for obtaining the segmentation of the first $\delta$ frames, we create copies of the initial optic flow.

Run-time: We implement our TMNet method in Pytorch on a NVidia TitanX GPU with 12GB memory for both training and testing. For our trained TMNet model, pre-processing steps like optic flow estimation and edge information extraction take around 0.07 s/frame and 0.05s/frame respectively. TMNet affinity prediction also takes 0.28 s/frame. Additionally, the clustering and tracking takes 2.17 s/frame. Speeding up the clustering process can be achieved by using the algorithm in [40]. To prove the speed up, we selected three random sequences from DAVIS16 dataset, applied our model that uses the algorithm in [40] for efficient clustering, and observed a great reduction in inference time. The average time of the clustering step reduced from 2.17 s/frame to 0.33 s/frame with a slight reduction in accuracy (of around 3%). There is often a trade-off between accuracy and computational speed when the range of additional used data is increased. In this case, the trade-off can be controlled by changing the number of neighbours used in the calculation of the affinity matrix.

4 Experimental results

4.1 Dataset and evaluation metrics

We report results on two widely used benchmarks: single object DAVIS16 dataset [46] and multi-object DAVIS17 dataset [43]. The datasets contain many challenging video sequences with multiple objects, occlusion, fast moving objects, background clutter, articulated motion, etc.

DAVIS16 [46] contains 50 HD video sequences, 3455 manual instance segmentation ground-truths. DAVIS17 [43] is a more challenging benchmark extending DAVIS16 [46] to multiple moving objects, and contains 120 HD video sequences (60 for train, 30 for val, 30 for test-dev) and 10K manual instance segmentation ground-truths. The task is more challenging due to the inclusion of multiple objects that additionally create occlusions, background clutter, etc.

We use the following performance measures described in DAVIS challenge [43] to evaluate the performance of our method:

- Region similarity metric ($J$) : Intersection over union between predicted and ground-truth segmentations.
- Contour accuracy metric ($F$) : Boundary accuracy of predicted boundaries against the ground-truth.
- Overall global metric ($J$ & $F$) : Average of $J$mean and $F$mean.
- Temporal decay metric ($T$) : Measure of consistency in labelling across the video sequence.

The $mean$ of a metric is the average error measured across all objects in all video sequences. The $recall$ is the fraction of sequences scoring higher than a threshold of 0.5.

4.2 Ablation study

To examine the effectiveness of the temporal neural attention and edge refinement components of our TMNet model individually, we performed an ablation study of our model on the DAVIS16 dataset [46].

The decrease in performance due to the removal of specific key components of our method is calculated as:

$$\Delta m = \left(1 - \frac{m_{\text{partial}}}{m_{\text{full}}}\right) \times 100\%$$

where $m$ represents the metric for which the decrease in performance is calculated ($J$ & $F$, $J$mean, $J$recall, $F$mean or $F$recall). $\Delta m$ represents the performance loss, $m_{\text{partial}}$ and $m_{\text{full}}$ represent the metric values without and with the specific component of our model whose efficiency is studied. Table 1 shows the results of our key component analysis.

The first row in Table 1 shows the loss in performance of our model without the temporal attention module, compared to the full model performance described in the third row. There is a significant improvement of 5.01% in global $J$ & $F$ metric due to the inclusion of the temporal attention.
This shows that the addition of tracking information helps the networks performance by resolving ambiguities in appearance of objects for challenging scenarios (unseen objects, dynamically varying number of objects, occlusions, non-rigid motions, and noisy background).

Similarly, the second row in Table 1 shows the loss in performance of our model without the edge refinement module. There is a 2.38% improvement due to the inclusion of the edge refinement module. It is also seen that edge refine module improves the boundary metric $F_{\text{mean}}$ by a large margin (3.52%). This performance gain is attributed to the edge refinement module as it improves the segmentation at object boundaries.

### 4.3 Evaluation results

#### 4.3.1 DAVIS16

We evaluated our method for UVOS on the single object DAVIS16 dataset [46]. Table 2 demonstrates that our method performs favourably compared to the state-of-the-art methods. Our method outperforms the state-of-the-art methods in one metric ($J_{\text{recall}}$), and the third best performance in both boundary metrics ($F_{\text{mean}}$ and $F_{\text{recall}}$) as highlighted in Table 2. The overall performance indicates the robustness of the appearance features, and is attributed to the temporal attention module. The increased performance in the boundary metrics compared to the other methods is attributed to the use of edge refinement module that refines object boundaries.

Figure 8 shows qualitative results of our method. The results for the sequence 'camel' shows the capability of our model to work on non-rigid articulated motions. In sequence 'blackswan', our method robustly segments the object amongst the noisy background. Apart from the quantitative results, our method also has another advantage. Different from most of the existing methods which perform binary foreground/background segmentation, our method is able to perform segmentation and tracking for multiple moving objects as explained in the next section.

#### 4.3.2 DAVIS17

To show that our method performs accurate segmentation for sequences with multiple moving objects, we applied our method to DAVIS17 dataset [43] and compared our results with existing methods. The performance of the proposed method is compared with several related state-of-the-art approaches (we have selected the top methods that do not use additional training data) including: (1) RVOS [23], (2) STEM-seg [51], (3) MATnet [17], (4) UnOVOST [11]. Table 3 compares the methods using the performance metrics described in the previous section.

The best results are obtained by UnOVOST [11]. This method is computationally expensive as it uses MaskRCNN [15] for object proposal generation. It also has many
heuristic post-processing steps requiring hyper-parameters tuning of multiple parameters (hyper-parameters are required for converting instance object mask proposals to short term tracklets, and merging short term tracklets to long term object trajectories), therefore limiting its use to specific datasets. Similarly MATnet [17] uses a CRF-based dataset specific heuristic post-processing to convert the foreground/background saliency maps, to multi-object segmentation. In contrast, our method uses no such post-processing, and shows comparable accuracy to other similar methods that perform multi-object segmentation directly without the requirement of any heuristic post-processing operations.

Overall, our method has the following advantages compared to the Mask-RCNN object detection and tracking methods [11, 23, 51]: (1) Our method performs category agnostic segmentation (segmentation of objects not seen by the training data) without using any prior knowledge of known object proposals from pre-trained ImageNet models (as we only use affinities to perform UVOS). (2) Our method has no heuristic post-processing steps that are required in Mask-RCNN based methods like multiple object proposal generation step, short term tracklet generation step, association step for long term tracklet generation and tracking (as we use correlation clustering on affinities learnt to perform UVOS). Hence our model can segment all probable moving object(s) and is capable of generalizing well to unseen object classes.

Figure 9 shows qualitative results for performing UVOS on multi-object sequences of DAVIS17 dataset [43]. The sequence ‘pigs’ demonstrates that the model can handle

### Table 3

Comparison with the state-of-the-art methods for unsupervised video object segmentation on multiple object DAVIS-17 dataset [43]

| Method      | J &F(Mean)↑ | J(mean)↑ | F(mean)↑ | F(decay)↓ | Heuristic post-processing |
|-------------|-------------|----------|----------|-----------|---------------------------|
| RVOS [23]   | 0.412       | 0.368    | 0.457    | 1.70      | No                        |
| STEm-Seg [51]| 0.647       | 0.615    | 0.678    | 1.20      | No                        |
| MATNet [17] | 0.586       | 0.567    | 0.604    | 1.80      | Yes                       |
| UnOVOST [11]| **0.679**   | **0.664**| **0.693**| **0.01**  | Yes                       |
| Target-Aware [52]| 0.650       | 0.637    | 0.662    | 9.4       | No                        |
| Ours        | 0.461       | 0.427    | 0.496    | 0.03      | No                        |

Bold signifies the best performing method

Our method, despite not using any post-processing, produces relatively accurate segmentation results.
multiple moving objects under challenging scenarios like occlusions and similar objects. Sequence 'dogs-jump' also shows the ability to handle category agnostic fast moving objects with similar appearance. The last two sequences of Fig. 9 shows qualitative results for performing UVOS on sequences of DAVIS17 dataset [43] consisting of several instances of objects belonging to the same category. The results demonstrate the ability of the method to handle occlusion by correctly segmenting only the object of interest that is moving (as we use temporal motion attentive features).
To show that our method is capable of generalizing well to unseen test videos, we train our model using the above mentioned DAVIS17 dataset [43] but test it on the FBMS dataset [53] (FBMS dataset is not used during training). FBMS dataset is composed of 59 sequences with sparse annotations and contains challenging multiple moving object sequences.

The performance of the proposed method on the test sequences of the FBMS dataset is compared with several related state-of-the-art approaches in Table 4. The results demonstrate that our method produces the second-best performance in the region similarity metric as highlighted in Table 4 (even when our method does not use the FBMS dataset for training). The sparseness of the groundtruth annotations does not affect our method compared to other methods (like LVO [24] & ARP [25]). Our method also produces multiple object segmentation different from other saliency-based methods that produce binary segmentation masks (like PDB [29] & COSNet [37]).

Figure 10 shows qualitative results for performing UVOS on challenging multiple object sequences of the FBMS dataset. The sequence ‘goats01’ demonstrates that the model can handle multiple moving objects under challenging scenarios like occlusions and similar objects. The results for the sequence ‘horses02’ shows the capability of our model to robustly segment very small objects amongst the noisy background that are very far away from the camera. This robustness is attributed to the temporal attention module that tries to segment anything that moves irrespective of the object class. This experiment also shows that our model does not overfit on the DAVIS16 and DAVIS17 datasets.

Table 4 Quantitative performance on the test sequences of FBMS dataset [53] using the Jmean (region similarity metric)

| Method     | SFL [47] | FSEG [48] | LVO [24] | ARP [25] | PDB [29] |
|------------|----------|-----------|----------|----------|----------|
| J mean†    | 56.0     | 68.4      | 65.1     | 59.8     | 74.0     |
| Method     | MATNet [17] | AGS [32] | COSNet [37] | FEMNet [39] | Ours    |
| J mean†    | 76.1     | 76.0      | 75.6     | 78.5     | 76.6     |

Bold highlights the metrics of the best and the second-best performing methods

Fig. 10 Qualitative results of our method for UVOS on the FBMS dataset. From top to bottom: goats01 sequence input and visual segmentation results, horses02 sequence input and visual segmentation results

4.3.3 FBMS

To show that our method is capable of generalizing well to unseen test videos, we train our model using the above mentioned DAVIS17 dataset [43] but test it on the FBMS dataset [53] (FBMS dataset is not used during training). FBMS dataset is composed of 59 sequences with sparse annotations and contains challenging multiple moving object sequences.

The performance of the proposed method on the test sequences of the FBMS dataset is compared with several related state-of-the-art approaches in Table 4. The results demonstrate that our method produces the second-best performance in the region similarity metric as highlighted in Table 4 (even when our method does not use the FBMS dataset for training). The sparseness of the groundtruth annotations does not affect our method compared to other methods (like LVO [24] & ARP [25]). Our method also produces multiple object segmentation different from other saliency-based methods that produce binary segmentation masks (like PDB [29] & COSNet [37]).

Figure 10 shows qualitative results for performing UVOS on challenging multiple object sequences of the FBMS dataset. The sequence ‘goats01’ demonstrates that the model can handle multiple moving objects under challenging scenarios like occlusions and similar objects. The results for the sequence ‘horses02’ shows the capability of our model to robustly segment very small objects amongst the noisy background that are very far away from the camera. This robustness is attributed to the temporal attention module that tries to segment anything that moves irrespective of the object class. This experiment also shows that our model does not overfit on the DAVIS16 and DAVIS17 datasets.
5 Conclusions

In this paper, we propose a new method (TMNet) to solve UVOS. Our model combines appearance, motion and edge cues. The motion cues from consecutive frames of video sequences help to find temporal connections, guide our model to learn powerful object representations, as they resolve ambiguities in appearance features through neural attention towards the moving object(s). The edge cues help to refine the errors at object boundaries where motion cues are inaccurate. Different from previous neural attention UVOS methods, our method predicts affinities instead of predicting binary segmentation masks, making the method capable of handling multiple moving objects in one forward pass. The model is efficiently optimized by a loss function on the predicted affinities using geometric constraints. Our experiments on two popular benchmarks (DAVIS16 and DAVIS17) demonstrate that TMNet is capable of effectively handling unseen object categories, multiple moving objects, occlusions, articulated object motions, and cluttered background. Extensive experiments on the datasets also show that the improvement in performance is due to the addition of the temporal neural attention and edge refinement modules.

Our method fails when objects in the video temporarily stop for multiple frames. This failure occurs since we process only selected number of frames at a time. So, we plan to extend the work further by storing important features of objects seen in a memory. This will allow the model to merge the tracklets accurately once the objects are re-identified later. Another area for improvement is to make use of the 3D scene flow available to incorporate the additional depth change information (not available in 2D optic flow) to aid the segmentation.

Acknowledgements This work was supported by the Australian Research Council through an ARC Linkage Project grant (LP160100662).

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. This work was supported by the Australian Research Council through an ARC Linkage Project grant (LP160100662).

Availability of data and materials Not applicable.

Code availability Not applicable.

Declarations

Conflict of interest Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

1. Hadizadeh H, Bajić IV (2013) Saliency-aware video compression. IEEE Trans Image Process 23(1):19–33
2. Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition, pp 3354–3361
3. Muthu S, Tennakoon R, Rathnayake T, Hoseinnezhad R, Suter D, Bab-Hadiashar A (2020) Motion segmentation of rgb-d sequences: combining semantic and motion information using statistical inference. IEEE Trans Image Process 29:5557–5570
4. Tennakoon R, Sadri A, Hoseinnezhad R, Bab-Hadiashar A (2018) Effective sampling: fast segmentation using robust geometric model fitting. IEEE Trans Image Process 27(9):4182–4194
5. Xu K, Wen L, Li G, Bo L, Huang Q (2019) Spatiotemporal cnn for video object segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1379–1388
6. Huang Y, Liu Q, Metaxas D (2009) Video object segmentation by hypergraph cut. In: IEEE conference on computer vision and pattern recognition, pp 1738–1745
7. Chang J, Fisher JW (2013) Topology-constrained layered tracking with latent flow. In: Proceedings of the IEEE international conference on computer vision, pp 161–168
8. Ranjan A, Jampani V, Balles L, Kim K, Sun D, Wulff J, Black MJ (2019) Competitive collaboration: joint unsupervised learning, camera motion, optical flow and motion segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 12240–12249
9. Tsai Y-H, Yang M-H, Black MJ (2016) Video segmentation via object flow. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3899–3908
10. Sun D, Roth S, Black MJ (2014) A quantitative analysis of current practices in optical flow estimation and the principles behind them. Int J Comput Vis 106(2):115–137
11. Zulfikar IE, Luiten J, Leibe B (2019) Unovost: unsupervised offline video object segmentation and tracking for the 2019 unsupervised davis challenge. In: Proceedings of the 2019 DAVIS challenge on video object segmentation-CVPR workshops, vol 3
12. Xu S, Liu D, Bao L, Liu W, Zhou P (2019) Mhp-vos: multiple hypotheses propagation for video object segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 314–323
13. Li F, Kim T, Humayun A, Tsai D, Rehg JM (2013) Video segmentation by tracking many figure-ground segments. In: Proceedings of the IEEE international conference on computer vision, pp 2192–2199

Ethics approval Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.
30. Lin H, Wu R, Liu S, Lu J, Jia J (2021) Video instance segmentation with transfer learning: within 5G infrastructure. Int J Mach Learn Cybern 12:3053–3067
31. Gowda SN, Eustratiadis P, Hospedales T, Sevilla-Lara L (2020) Stochastic selection with a propose-reduce paradigm. arXiv preprint arXiv:2103.13746
32. Yang Z, He X, Gao J, Deng L, Smola A (2016) Stacked attention networks for image question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 21–29
33. Abdal M, Ahmad A, Jeon G (2020) Top view multiple people tracking by detection using deep sort and yolov3. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
34. Zhou Z, Zhang W, Zhao J (2019) Robust visual tracking using discriminative sparse collaborative map. Int J Mach Learn Cybern 10(11):3201–3212
35. Zhuang D, Jiang M, Kong J, Liu T (2021) Spatiotemporal attention enhanced features fusion network for action recognition. Int J Mach Learn Cybern 12(6):1583–1596
36. Wang W, Sun H, Zhao S, Tao R, Shen J, Jia Q, Wang Q, Bertinetto L, Shen J, Zhao S, Hoi SC, Ling H (2019) Learning unsupervised video object segmentation through visual attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3064–3074
37. Zhou Y, Xu X, Shen F, Zhu X, Shen HT (2021) Flow-edge guided unsupervised video object segmentation. IEEE Transactions on circuits and systems for video technology
38. Keuper M, Levinkov E, Bonneel N, Lavoué G, Brox T, Andres B (2015) Efficient decomposition of image and mesh graphs by lifted multicuts. In: Proceedings of the IEEE international conference on computer vision, pp 1751–1759
39. Pont-Tuset J, Perazzi F, Caelles S, Arbeláez P, Sorkine-Hornung A, Van Gool L (2017) The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675
40. Cao W, Wang X, Ming Z, Gao J (2018) A review on neural networks for image question answering. In: Proceedings of the IEEE/CVF international conference on computer vision and pattern recognition, pp 3075–3084
41. Jain SD, Xiong B, Grauman K (2017) Fusionseg: learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 91–99
42. Zhou Y, Xu X, Shen F, Zhu X, Shen HT (2016) Stacked attention networks with random weights. Neurocomputing 275:278–287
43. Keuper M, Andres B, Van Gool L (2017) The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675
44. Athar A, Mahadevan S, Osep A, Leal-Taixé L, Leibe B (2020) Making a case for 3d convolutions for object segmentation in videos. arXiv preprint arXiv:2008.11516
45. Mahadevan S, Athar A, Osep A, Hennen S, Leal-Taixé L, Leibe B (2019) Anchor diffusion for unsupervised video object segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 931–940
46. Abdal M, Ahmad A, Jeon G (2020) Top view multiple people tracking by detection. Philos Trans R Soc Lond Ser A: Math Phys Eng Sci 356(1740):1321–1340
47. Bideau P, Learned-Miller E (2016) A detailed rubric for motion segmentation. arXiv preprint arXiv:1610.00334
48. Lu X, Qi Y, Wang Z, Chen K, Liu Z, Shi J, Luo P, Tang X, Loy CC (2017) Video object segmentation with re-identification. arXiv preprint arXiv:1708.00197
49. Keuper M, Levinkov E, Bonneel N, Lavoué G, Brox T, Andres B (2015) Efficient decomposition of image and mesh graphs by lifted multicuts. In: Proceedings of the IEEE international conference on computer vision, pp 1751–1759
50. Pont-Tuset J, Perazzi F, McWilliams B, Van Gool L, Gross M, Sorkine-Hornung A (2016) A benchmark dataset and evaluation methodology for video object segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 724–732
51. Keuper M, Andres B, Brox T (2015) Motion trajectory segmentation via minimum cost multicuts. In: Proceedings of the IEEE international conference on computer vision, pp 3271–3279
52. Jain SD, Xiong B, Grauman K (2017) Fusionseg: learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 2117–2126
53. Yang Z, Wang Q, Bertinetto L, Hu W, Bai S, Torr PH (2019) Anchor diffusion for unsupervised video object segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 931–940
54. Abdal M, Ahmad A, Jeon G (2020) Top view multiple people tracking by detection using deep sort and yolov3. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
55. Zhou Z, Zhang W, Zhao J (2019) Robust visual tracking using discriminative sparse collaborative map. Int J Mach Learn Cybern 10(11):3201–3212
56. Zhu T, Li J, Wang S, Tao R, Shen J (2020) Matnet: motion-attentive transition network for zero-shot video object segmentation. IEEE Trans Image Process 29:8326–8338
57. Spelke ES (1990) Principles of object perception. Cogn Sci 14(1):29–56
58. Koffka K (2013) Principles of gestalt psychology. Routledge, London
59. Torr PH (1998) Geometric motion segmentation and model selection. Philos Trans R Soc Lond Ser A: Math Phys Eng Sci 356(1740):1321–1340
60. Bideau P, Learned-Miller E (2016) A detailed rubric for motion segmentation. arXiv preprint arXiv:1610.00334
61. Lu X, Qi Y, Wang Z, Chen K, Liu Z, Shi J, Luo P, Tang X, Loy CC (2017) Video object segmentation with re-identification. arXiv preprint arXiv:1708.00197
62. Keuper M, Levinkov E, Bonneel N, Lavoué G, Brox T, Andres B (2015) Efficient decomposition of image and mesh graphs by lifted multicuts. In: Proceedings of the IEEE international conference on computer vision, pp 1751–1759
63. Pont-Tuset J, Perazzi F, Caelles S, Arbeláez P, Sorkine-Hornung A, Van Gool L (2017) The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675
64. Cao W, Wang X, Ming Z, Gao J (2018) A review on neural networks for image question answering. In: Proceedings of the IEEE/CVF international conference on computer vision and pattern recognition, pp 3075–3084
65. Zhou Y, Xu X, Shen F, Zhu X, Shen HT (2021) Flow-edge guided unsupervised video object segmentation. IEEE Transactions on circuits and systems for video technology
66. Keuper M, Levinkov E, Bonneel N, Lavoué G, Brox T, Andres B (2015) Efficient decomposition of image and mesh graphs by lifted multicuts. In: Proceedings of the IEEE international conference on computer vision, pp 1751–1759
67. Pont-Tuset J, Perazzi F, McWilliams B, Van Gool L, Gross M, Sorkine-Hornung A (2016) A benchmark dataset and evaluation methodology for video object segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 724–732
68. Keuper M, Andres B, Brox T (2015) Motion trajectory segmentation via minimum cost multicuts. In: Proceedings of the IEEE international conference on computer vision, pp 3271–3279
69. Jain SD, Xiong B, Grauman K (2017) Fusionseg: learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 2117–2126
70. Yang Z, Wang Q, Bertinetto L, Hu W, Bai S, Torr PH (2019) Anchor diffusion for unsupervised video object segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 931–940
71. Mahadevan S, Athar A, Osep A, Hennen S, Leal-Taixé L, Leibe B (2020) Making a case for 3d convolutions for object segmentation in videos. arXiv preprint arXiv:2008.11516
72. Athar A, Mahadevan S, Osep A, Leal-Taixé L, Leibe B (2020) Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In: European conference on computer vision, pp 158–177
73. Zhou T, Li J, Li X, Shao L (2021) Target-aware object discovery and association for unsupervised video multi-object segmentation.
In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6985–6994

53. Brox T, Malik J (2010) Object segmentation by long term analysis of point trajectories. In: European conference on computer vision, pp 282–295

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.