INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

Segmentation of Moving Objects

Final Report

COURSE CODE: Computer Vision - CS7.501.S22

Advisor:
Prof. Anoop Namboodiri

Team Name:
Stochastic Squad

Project Mentor:
Abhishek G

Project Representatives:
Aditya Kumar Singh - 2021701010
Dhruv Srivastava - 2021701021
Rohit Girmaji - 2021900013

Academic year:
2021-2022

Contents

1. Project Description

1.1. Problem statement

The task of Moving Objects Segmentation also called Motion Segmentation (or Video Object Segmentation - **VOS**) aims at segmenting objects in a video that exhibit independent motion in at least one frame. The Moving objects/pixels belong to foreground and the remaining pixels are the background. This task has many practical applications in understanding scenes for Autonomous driving, virtual background creation, robotics, automated surveillance, social media, augmented reality, movie production, video conferencing, etc.
This problem is dealt in many ways in the literature and broadly categorized into methods based on training and architectures used in training. Based on the training methods (i.e., how human interventions are involved), VOS methods are grouped into 4 types [15]: *Unsupervised* (**UVOS** or **AVOS**[1]), *Semi-supervised* (**SVOS**[2]), *Interactive* (**IVOS**) and *Referring* (language guided, also called **LVOS**). Based on the architectures, it is grouped into 3: *Deep Learning Techniques*(CNN's, Transformers, Encoder-Decoder etc), *Non-Deep Learning Techniques* (frame differences, optical flow, MRF/CRF) and *combination of both*. In this project we deal with unsupervised and semi-supervised methods using Deep Learning and Non-Deep Learning based techniques.

Problem Formulation and Taxonomy
Formally, let $\mathcal{X}$ and $\mathcal{Y}$ denote the input space and output segmentation space, respectively. Deep learning-based video segmentation solutions generally seek to learn an ideal video-to-segment mapping $f^* : \mathcal{X} \longrightarrow \mathcal{Y}$.

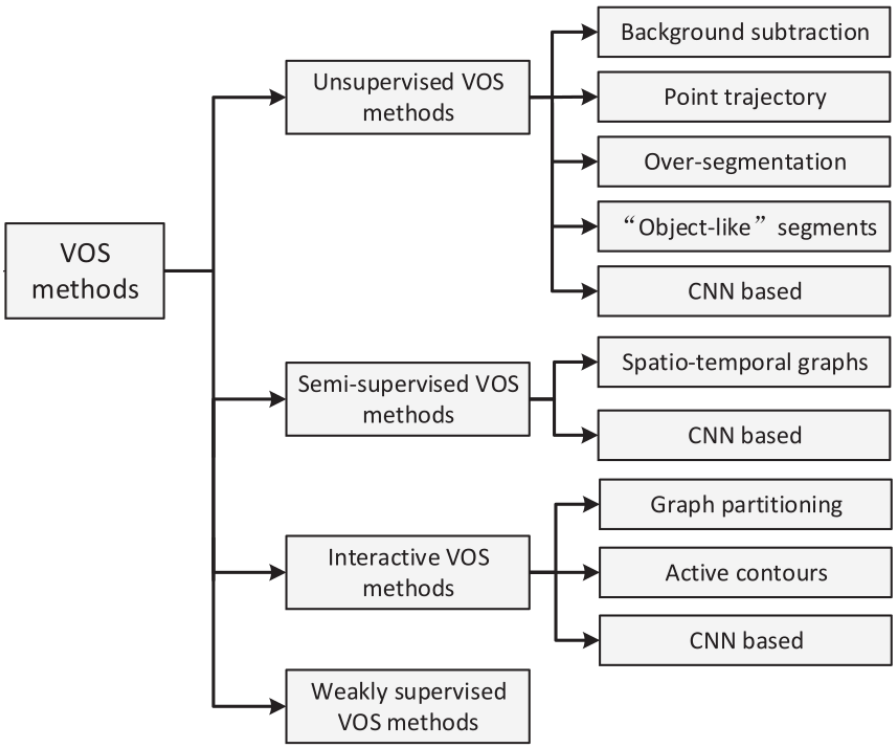

Figure 1: Taxonomy of video object segmentation[17]

[1] AVOS: Automatic Video Object Segmentation
[2] Also called Semi-automatic Video Object Segmentation in some literature.

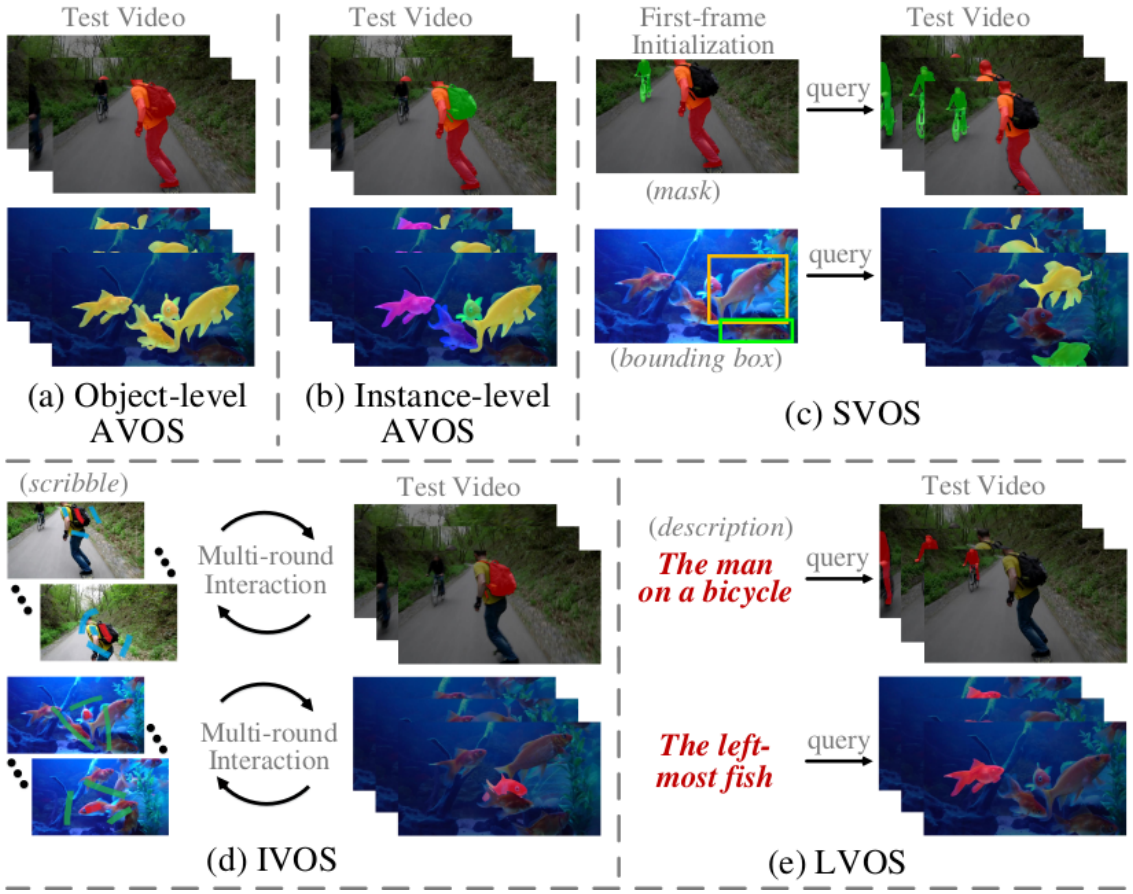

Figure 2: Video segmentation tasks reviewed: (a) object-level automatic video object segmentation (object-level AVOS), (b) instance-level automatic video object segmentation (instance-level AVOS), (c) semi-automatic video object segmentation (SVOS), (d) interactive video object segmentation (IVOS), (e) language-guided video object segmentation (LVOS)[15]

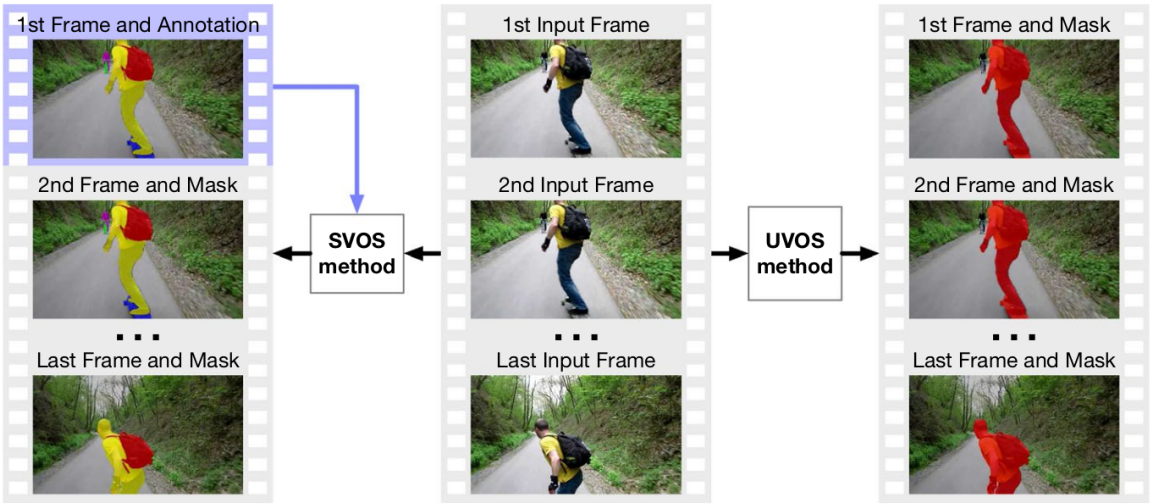

Figure 3: Diagram of UVOS and SVOS methods. UVOS methods segment the objects with dominant movement or visual saliency. In contrast, the target objects (the ones to segment) in SVOS depend on the human annotations in the first frame (highlighted in purple). Therefore, SVOS methods have more flexibility in defining target objects[8]

A brief chronology of the VOS methods, where some milestone works from 2006 to 2021 are highlighted.

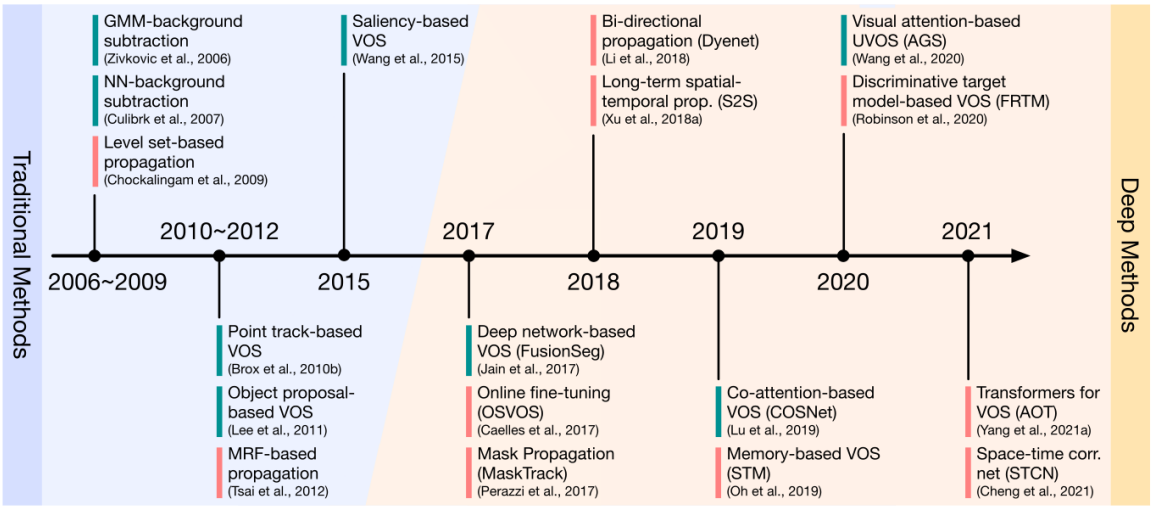

Figure 4: Blue-marks: Unsupervised VOS methods; Red marks: semi-supervised VOS methods.[8]

1.2. Overview of Approaches

1. **UVOS**: These methods (Fig 2(a-b)) aim to automatically segment primary objects without any user annotations. Without knowing the foreground objects in advance, finding the most prominent objects in a video frame is a challenging task. That is, the UVOS not only solves SVOS (e.g., occlusion, object deformation, and background clutters), but also how to distinguish the primary objects from complex and diverse backgrounds.
2. **SVOS**: *Semi-supervised* or *One-shot learning* (Fig 2(c)) utilizes a single labeled frame (usually the first frame of the video) to estimate the remaining frame segmentation in the video. In contrast, SVOS methods reach better performance than unsupervised ones, generally. The object handled by SVOS task is arbitrary, and no further assumptions are made on the object class. Therefore, this task still has great challenges, because the appearance of objects changes over time, such as occlusion, motion blur, fast motions, scale variations of different objects, and more.
3. **IVOS**: From the perspective of user interaction with the methods, there are two extremes: The unsupervised scenarios do not require any user input, while SVOS requires precise pixel-level segmentation in the first frame. IVOS methods (Fig 2(d)) provide a more efficient alternative by incorporating user guidance throughout the analysis process, where unsupervised methods and semi- methods collaborate.
4. **LVOS**: Language-guided video object segmentation (LVOS) (Fig 2(e)) is a sub-branch of SVOS, where human intervention is given as linguistic descriptions about the desired objects, enabling efficient human-computer interaction. This method has to meet two requirements: (i) correction with minimal user interaction (such as clicks and scribbles) and (ii) understanding the user's intentions.

In our project we'll be experimenting with **two** deep-learning based UVOS methods, namely, *MATNet* and *RTNet*, and **one** deep-learning based SVOS method involving Transformers (*TransVOS*). As a check for baselines, we will leverage upon the *optical flow* as well as *background-subtraction* techniques to segment the moving objects from background.

1.3. Overview of upcoming sections

2. Datasets

To evaluate the performance of various video object segmentation and tracking methods, one needs test video dataset, the ground truth. In this section, we will give a brief introduction of datasets.

2.1. Davis 16, 17 and 18

DAVIS 2016, 2017, and 2018 datasets[3] are some of the most popular datasets for training and evaluating video object segmentation algorithms. DAVIS 2016 [10] dataset contains 50 full high quality video sequences with

[3] https://davischallenge.org/

3,455 annotated frames in total, and focuses on single-object video object segmentation, that is, there is only one foreground object per video. 30 training set and 20 validation set in this dataset is divided. Later, DAVIS 2017 [12] dataset complements DAVIS 2016 dataset training and validation sets with 30 and 10 high-quality videos, respectively. It also provides an additional 30 development test sequences and 30 challenge test sequences. Also, the DAVIS 2017 dataset relabels multiple objects in all video sequences. These improvements make it more challenging than the original DAVIS 2016 dataset. In addition, Each video is labeled with multiple attributes such as occlusion, object deformation, fast motion, and scale change to provide a comprehensive analysis of model performance. Moreover, DAVIS 2018 dataset [4] adds 100 videos with multiple objects per video to the original DAVIS 2016 dataset and complements an interactive segmentation teaser track.

For our use case we used DAVIS2017 dataset only.

2.2. FBMS

FBMS-59[4] (Freiburg-Berkeley motion segmentation) [9] are widely used for unsupervised and semi-supervised VOS methods. FBMS-59 is an extension of BMS-26 [2] dataset that consists 26 videos with a total of 189 annotated image frames, with shots from movie stories and the 10 vehicles and 2 human sequences. The FBMS-59 dataset reflects two major improvements in the previous version of BMS-26. *First*, the updated version dataset adds 33 new sequences; therefore, the FBMS-59 dataset consists of 59 sequences. *Second*, these 33 new sequences incorporate challenges of unconstrained videos such as fast motion, motion blur, occlusions, and object appearance changes. The sequences are divided into 29 training and 30 test video sequences.

2.3. YoutubeVOS

This is a large-scale dataset[5] series for VOS, with long-range video sequences; it contains three versions: YouTube-VOS 2018, YouTube-VOS 2019, and YouTube-VIS [16]. The first two versions are designed for multi-object SVOS, while the latter one serves for multi-object UVOS. From Table 1, it can be found that the number of video sequences in YouTube-VOS is dozens of times as many as that in DAVIS, which indicates more diverse objects and context are considered. Moreover, each video sequence in the datasets has a greater number of frames than any other datasets, allowing VOS methods to model and exploit long-range temporal dependency between frames. Because the amount of the data is huge, the YouTub-VOS team only managed to provide the pixel-level object masks for every 5^{th} frame. To better validate the generalisation ability of VOS models, YouTube-VOS groups the contained object categories into two sets: 'seen' and 'unseen', where the objects belonging to 'unseen' categories only residents in the testing set, and the ones belonging to 'seen' categories residents in both training and testing sets. By comparing the segmentation results on 'seen' and 'unseen' objects, the performance of VOS models on generalisation can be evaluated. To echo DAVIS, the YouTube-VOS team organises a challenge[5] on VOS annually since 2018.

2.4. Self captured videos

Here we will test upon some real-world videos captured through our mobile phones. Their results and evaluations is mentioned in section 4.

3. Project Implementations

3.1. Background creation

3.2. Frame by frame subtraction

3.3. Optical flow in greyscale thresholded

[4] https://lmb.informatik.uni-freiburg.de/resources/datasets/
[5] https://youtube-vos.org/

3.4. MATNet

Inspired by the human vision system (HVS), which has remarkable motion perception capabilities to quickly orient human attention to moving objects in dynamic scenes, *MATNet* leverages motion cues as bottom-up signal to guide the perception of object appearance. To achieve this, an asymmetric attention block, named Motion-Attentive Transition (MAT), is proposed within a two-stream encoder network to firstly identify moving regions and then attend appearance learning to capture the full extent of objects. Putting MATs in different convolutional layers, the encoder becomes deeply interleaved, allowing for close hierarchical interactions between object appearance and motion. Such a biologically-inspired design is proven to be superb to conventional two-stream structures, which treat motion and appearance independently in separate streams and often suffer severe overfitting to object appearance.

Even before the era of deep learning, *object motion* has always been considered as one of the most important cues for automatic video object segmentation which has been expoited heavily by early non-learning methods. These approaches are built upon handcrafted features (e.g., motion boundary, saliency, point trajectories) and rely heavily on classic heuristics in video segmentation (e.g., object proposal ranking, spatiotemporal coherency, long-term trajectory clustering). Although these methods can work in a purely unsupervised way, they suffer the limited representability of the handcrafted features. More recently, research has turned towards the deep learning paradigm, with several studies from which one way to cast this problem is to having a zero-shot solution (ZVOS) [14] [5]. This is different from one-shot video object segmentation (OVOS) [3], which requires first-frame annotations for model adaption to test data in the inference phase.

3.4.1 Model Overview

1. A novel deeply interleaved two-stream network architecture has been proposed to learn powerful spatio-temporal object representations for *ZVOS* (Zero-Shot Video Object Segmentation). This is achieved by an asymmetric attention module, i.e., MAT, that accounts for object motion and appearance interactions in a more comprehensive way.
2. Second, a *boundary-aware decoder* is introduced to obtain segmentation with crisp object boundaries. The decoder learned with a novel adapted cross-entropy loss produces accurate boundaries in regions of primary objects.
3. Based on these designs, MATNet consistently outperforms state-of-the-art methods over several ZVOS benchmarks.

The proposed MATNet is designed as a unified of three tightly coupled sub-networks: *Interleaved Encoder Network*, *Bridge Network* and *Boundary-Aware Decoder Network*. The pipeline is illustrated in Fig 5

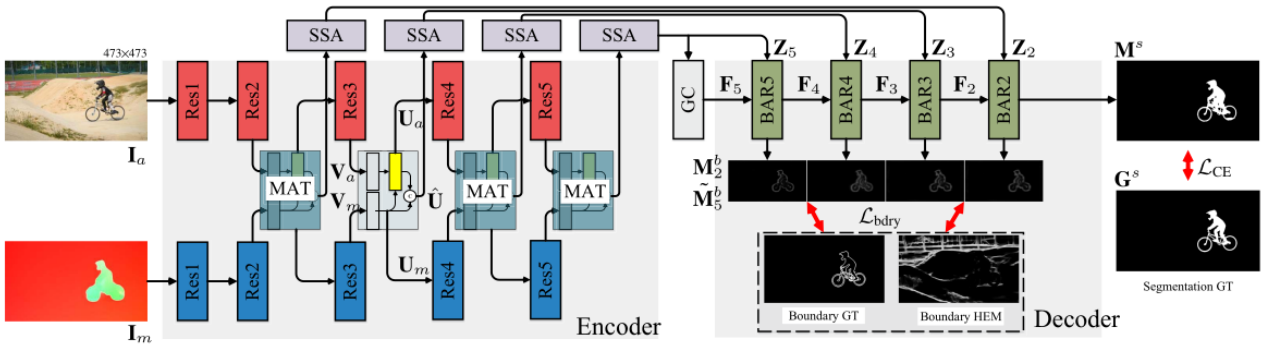

Figure 5: Matnet

3.4.2 Interleaved Encoder Network

The encoder jointly captures the spatiotemporal information using a two-stream structure that incorporates a **MAT** module (refer 6) into each network layer, which offers a motion-to-appearance pathway for information exchange. More technically, we take the first 5 convolutional blocks of ResNet-101 as the backbone for each stream. Given an RGB frame $\mathbf{I}_a \in \mathbb{R}^{w\times h\times 3}$ and its optical flow field $\mathbb{I}_m \in \mathbb{R}^{w\times h\times 3}$, the encoder first extracts intermediate appearance and motion features separately at the $i-$th ($i \in \{2,3,4,5\}$) residual stage, denoted as $\mathbf{V}_{a,i} \in \mathbb{R}^{W\times H\times C}$ and $\mathbf{V}_{m,i} \in \mathbb{R}^{W\times H\times C}$, where W, H, and C represent the spatial width, height and channel number of the feature tensors, respectively. The features are subsequently enhanced by a MAT module $\mathcal{F}_{\text{MAT}}$

as:

$$\hat{\mathbf{U}}_{a,i}, \hat{\mathbf{U}}_{m,i} = \mathcal{F}_{\text{MAT}}\left(\mathbf{V}_{a,i}, \mathbf{V}_{m,i}\right) \tag{1}$$

where $\hat{\mathbf{U}}_{\cdot,i} \in \mathbb{R}^{W\times H\times C}$ represents the enriched features. For the $i-$th stage, the spatiotemporal object representation $\hat{\mathbf{U}}_i$ is obtained as $\hat{\mathbf{U}}_i = concat\left(\hat{\mathbf{U}}_{a,i}, \hat{\mathbf{U}}_{m,i}\right) \in \mathbb{R}^{W\times H\times 2C}$ which is further fed into the down-stream decoder via a bridge network.

MAT - Motion-Attentive Transition Module

Each MAT module is comprised of two soft attention units and one attention transition unit, as depicted in Fig. 6. The soft attention units help to emphasize the most informative regions in the appearance or motion feature maps, while the transition unit transfers the attentive motion features to facilitate spatiotemporal feature learning.

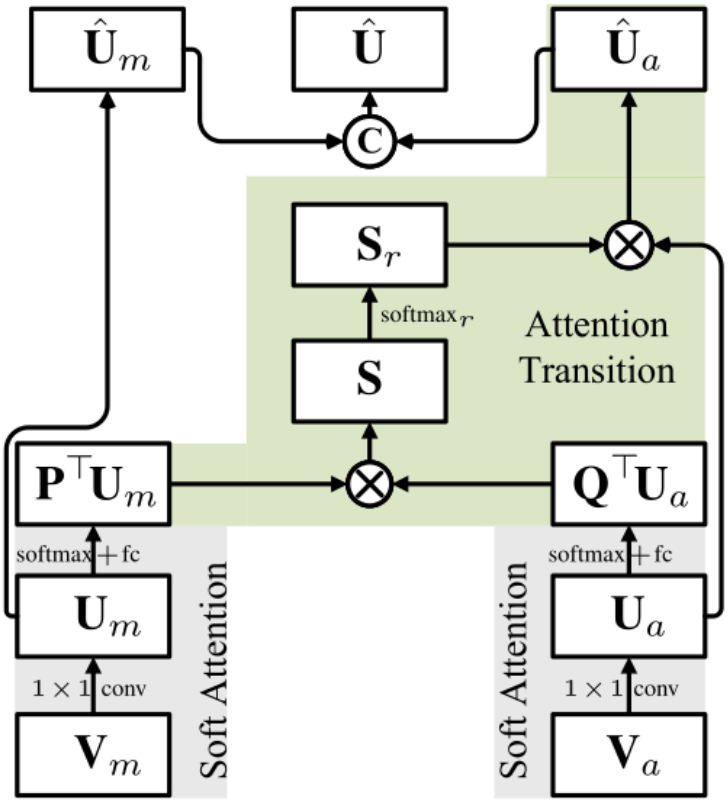

Figure 6: Computational graph of MAT. $\otimes$ and Ⓒ indicate matrix multiplication and concatenation operations, respectively.

To know intricacies of this module refer to [18].

3.4.3 Bridge Network

The bridge network is responsible for selecting informative spatiotemporal features for the decoder. It is built upon several SSA modules (refer to the following part), each of which takes advantage of U_i at the $i-$th stage, attending it both locally and globally to produce attentive feature Z_i, with a unified attention module. The local attention adopts channel-wise and spatial-wise attention mechanisms to highlight the correct object regions and suppress possible noise existing in the redundant features, while the global attention aims to re-calibrate the features to account for objects of different sizes.

SSA - Scale-Sensitive Attention Module

The SSA module $\mathcal{F}_{\text{SSA}}$ is extended from a simplified CBAM $\mathcal{F}_{\text{CBAM}}$ by adding a global attention F_g . Given a feature map $\mathbf{U} \in \mathbb{R}^{W\times H\times 2C}$, our SSA module refines it as follows:

$$\mathbf{Z} = \mathcal{F}_{\text{SSA}} = \mathcal{F}_g\left(\mathcal{F}_{\text{CBAM}}(\mathbf{U})\right) \in \mathbb{R}^{W\times H\times 2C} \tag{2}$$

The CBAM module $\mathcal{F}_{\text{CBAM}}$ consists of two sequential sub-modules: *channel* and *spatial attention*, which can be formulated as:

$$\begin{aligned} \textit{Channel Attention}\text{: } \mathbf{s} = \mathcal{F}_s(\mathbf{U}),\ \mathbf{e} = \mathcal{F}_e(\mathbf{s}),\ \mathbf{Z}_c = \mathbf{e} \star \mathbf{U}, \\ \textit{Spatial Attention}\text{: } \mathbf{p} = \mathcal{F}_p(\mathbf{Z}_c),\ \mathbf{Z}_{\text{CBAM}} = \mathbf{p} \odot \mathbf{Z}_c \end{aligned} \tag{3}$$

where $\mathcal{F}_s$ is a squeeze operator that gathers the global spatial information of $\mathbf{U}$ into a vector $\mathbf{s} \in \mathbb{R}^{2C}$, while $\mathcal{F}_e$ is an excitation operator that captures channel-wise dependencies and outputs an attention vector $\mathbf{e} \in \mathbb{R}^{2C}$. For more info please refer to [18].

3.4.4 Boundary-Aware Decoder Network

The decoder network adopts a coarse-to-fine scheme to conduct segmentation inference. It consists of four *BAR* modules (refer to 3.4.4), i.e., $\mathcal{F}_{\text{BAR}_i}$, $i \in \{2, 3, 4, 5\}$, each corresponding to the i-th residual block. From $\mathcal{F}_{\text{BAR}_5}$ to $\mathcal{F}_{\text{BAR}_2}$, the resolution of feature maps gradually increases by compensating for high-level coarse features with more low-level details. The $\mathcal{F}_{\text{BAR}_2}$ produces the finest feature maps, whose resolutions are 1/4 of the input image size. They are sequentially processed by three additional layers, i.e., conv$(3 \times 3, 1)$, upsampling and sigmoid, to obtain the final mask output $\mathbf{M}^s \in \mathbb{R}^{w \times h}$.

BAR - Boundary-Aware Refinement Module

In the decoder network, each BAR $\mathcal{F}_{\text{BAR}_i}$ accepts two inputs, Z_i from the corresponding SSA module and F_i from the previous BAR. To obtain a sharp mask output, the BAR first performs object boundary estimation using an extra boundary detection module $\mathcal{F}_{\text{BDRY}}$, which compels the net-work to emphasize finer object details. The predicted boundary map is then combined with the two inputs to produce finer features for the next BAR module. This can be formulated as:

$$\begin{aligned} \mathbf{M}_i^b &= \mathcal{F}_{\text{BDRY}}(\mathbf{F}_i), \\ \mathbf{F}_{i-1} &= \mathcal{F}_{\text{BAR}_i}(\mathbf{Z}_i, \mathbf{F}_i, \mathbf{M}_i^b) \end{aligned} \tag{4}$$

where $\mathcal{F}_{\text{BDRY}}$ consists of a stack of convolutional layers and a sigmoid layer (see Fig. 7), $\mathbf{M}_i^b \in \mathbb{R}^{w \times h}$ indicates the boundary map and $\mathbf{F}_{i-1}$ is the output feature map of BAR_i.

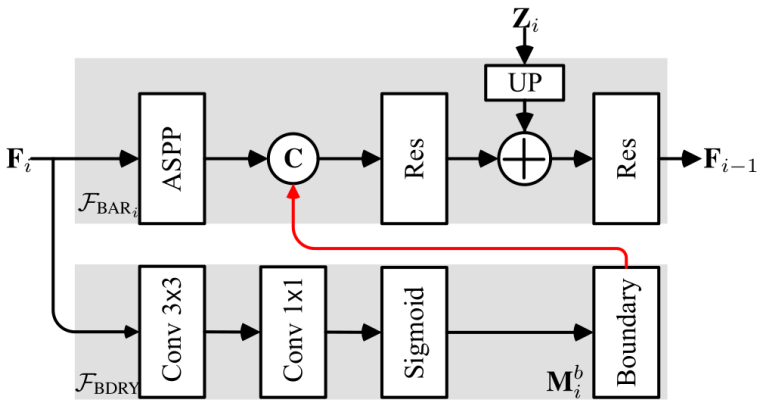

Figure 7: Computational graph of the BAR i module. Here, 'Res' is a residual block, while 'UP' denotes bilinear upsampling. ⓒand ⊕ indicate concatenation and element-wise addition operations, respectively.

3.5. RTNet

General methods for video object segmentation treat moving objects as primary objects and rely on optical flow to capture motion cues in videos. However, flow information alone is insufficient to distinguish primary objects from background objects that move together because both features interfere and the primary object's appearance is misguided.
The proposed solution against these issues is a Reciprocal Transformation Network RTNet which inherently uses a Reciprocal Transform Module (RTM) and Spatial Temporal Attentive Fusion Module (STAFM) details about which are shared in subsequent sections.

3.5.1 Reciprocal Transformation Network (RTNet)

RTNet discovers the primary objects by correlating three key factors: the intra-frame contrast, the motion cues, and temporal coherence of recurring objects. It is claimed that this enhances the motion features in order to focus on moving objects with salient appearance while removing co-moving outliers.

3.5.2 Reciprocal Transform Module (RTM)

RTM allows in-domain and cross-domain feature interactions. This calculates similarities for all pairwise features, such as motion-motion, appearance-appearance, and appearance-motion. When RTM is used, it produces three different sorts of primary object properties; 1. Self-similarities of appearance and motion features, which lead to intra-frame contrast; 2. appearance-motion similarity which reflects motion cues; 3. cross-frame appearance-appearance and motion-motion features similarities which yield temporal coherence.
RTM consists of three sub-modules: reciprocal scaling, reciprocal transformation, and reciprocal gating.

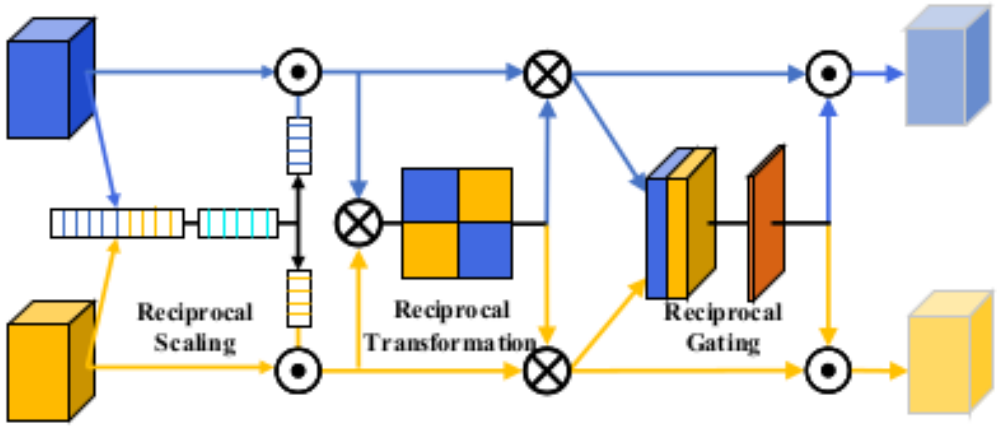

Figure 8: Reciprocal Transform Module (RTM)

3.5.3 Spatial Temporal Attentive Fusion Module (STAFM)

For focusing on global information while preserving the local context per frame, a STAFM is used to leverage the appearance and motion features from the corresponding encoder state, and segment spatio-temporally consistent primary objects.

3.5.4 Model Design

Given a pair of frames Ia1, Ia2 in a video and their corresponding optical flow Of_fw and Of_bw computes using RAFT, the aim is to segment the primary objects within both the frames. The flow of frames is divided into two parallel pipelines, an appearance stream which takes in RGB frames Ia1 and Ia2 as inputs and other is the motion stream Of_fw and Of_bw as inputs. Both the streams adopt a ResNet backbone with dilated convolutions- an encoder-decoder architecture with skip connections. At every stage of encoder, RTM is applied to mutually evolve and integrate the pairwise features. The decoder stage of the

3.6. TransVOS

TransVOS is a transformer based architecture for Semi-Supervised Video Object Segmentation. It takes a query frame (current frame) and reference sets (history frames with predicted masks) as input and outputs segmentation map of the query frame. The architecture mainly consists of 3 parts: Feature Embedding (Feature Extractor), Relationship Modelling (Vision Transformer), Segmentation each of which are described below.

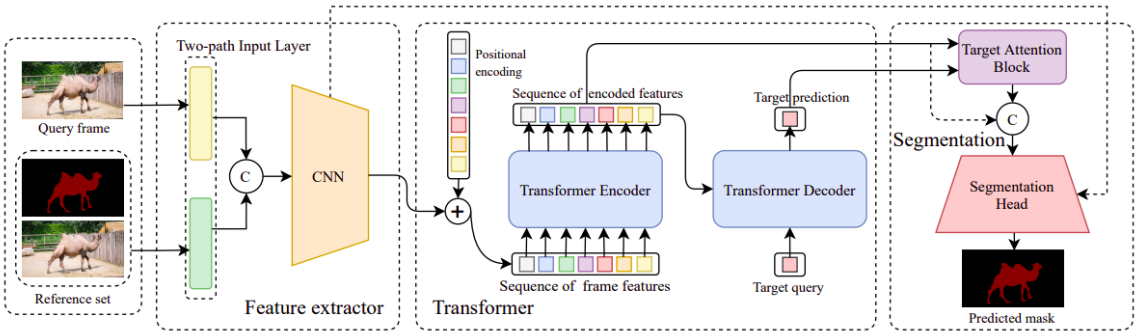

Figure 9: TransVOS Architecture for Semi-Supervised Video Object Segmentation

3.6.1 Feature Embedding (Feature Extractor)

To exploit the spatial and temporal information present in the query frame and reference sets this feature extractor is designed to extract features of the target object that can be mapped to a embedding space. It consists of two-path input layer and a bottle neck layer to reduce the channel dimension of the output from two-path input layer. The two-path input layer is designed to encode two types of inputs i.e, RGB frame(query frame) and RGB frames with corresponding object masks(reference sets). One path contains a convolution operation to encode the input RGB frame. The other path contains 3 convolution operations to encode the RGB frame,Object mask's foreground and background respectively. Four stages of ResNet is used as the convolutional network to perform the encoding. The output from these two paths is concatenated along the temporal dimension and then fed into a bottleneck layer to reduce the channel dimension. This output is flattened into one dimension which serves as the input to the vision transformer.

3.6.2 Relationship Modelling (Vision Transformer)

Vision Transformer tries to model the spatial temporal relationships using the feature embedding obtained from the previous layer(Feature extractor). It consists of Transformer Encoder and Decoder. Encoder tries to get the correspondence among the input frames pixels with the help of Positional encoding which contains information about space-time position. The encoder could learn the target object's correspondence among the input frames and model target object structure in a specific frame. Then the decoder can predict the spatial position of the target objects in the query frame and get robust representations for the target object.

3.6.2.1 Positional Encoding

The Sinusoidal position Encoding PE is added to the features X from feature extractor to get the final input Z to the transformer.

$$Z = X + PE, \tag{5}$$

where $PE(pos, 2i) = \sin(\frac{pos}{10000^{\frac{2i}{d}}})$, $PE(pos, 2i+1) = \cos(\frac{pos}{10000^{\frac{2i}{d}}})$

3.6.2.2 Transformer Encoder

Transformer Encoder is used to model the spatio temporal relationships among pixel level features of the sequence. It takes features Z and outputs encoded features E. The encoder consists of N layers each of which have Multi Head Self attention, feed forward layer.Multi Head self attention module is used to capture the spatio-temporal relationships from different representation sub-spaces.

3.6.2.3 Transformer Decoder

The decoder tries to predict the spatial position of the target object in query frame using the encoded features E and target query x_o and output decoded features O.The decoder consists of N layers each of which consists of Multi Head Self attention,multi-head cross attention, feed forward layer. Multi-head cross attention layer is leveraged to fuse target object features from the encoder. multi-head self attention is used to integrate target object information from different representation sub space.

3.6.3 Segmentation

3.6.3.1 Target attention block

To get the target mask prediction from the transformer output the model needs the target's mask features of the query frame. For this Target Attention Block takes the encoder query frame features and decoder output and finds it similarity by passing them to a multi head self attention block. The output from this block are attention maps which are concatenated with the encoder features and fed to the segmentation head as input S.

3.6.3.2 Segmentation Head

This layer takes the above obtained features S as the input to output the predicted masks. The current frame feature maps obtained from the feature extractor along with the features S are input to the two blocks present in this layer. Skip connections are made between both the blocks. Gradual upscaling is done by a factor and then a convolution and softmax operation is performed to get the final predicted mask at 1/4th of the original image size. Then bilinear interpolation is done to upscale the image to the size of original image(query image).

4. Results

4.1. Evaluation Metrics

Presently, three metrics are frequently used [11] to measure how object-level AVOS methods perform on this task, where both the predicted results and the ground truth are used for foreground-background partitioning. Measures can be focused on evaluating which pixels of the ground truth are detected, or indicating the precision of the bounding box.

- **Region Jaccard or Region Similarity** $\mathcal{J}$: The region similarity is the Intersection over Union (IoU) function between the predicted object segmentation mask M and ground truth G. This quantitative metric is used for measuring the number of misclassified image pixels and measuring pixels matching segmentation algorithm. It is calculated between the segmentation results $\hat{Y} \in \{0,1\}^{w \times h}$ and the ground-truth $Y \in \{0,1\}^{w \times h}$:

$$\mathcal{J} = \frac{\hat{Y} \cap Y}{\hat{Y} \cup Y} \tag{6}$$

  which computes the number of pixels of the intersection between $\hat{Y}$ and Y, and divides it by the size of the union.
- **Contour Precision** $\mathcal{F}$: The segmented mask is treated as a set of closed contour regions, and the function of precision (P_c) and recall (R_c) is to calculate the *contour-based F-measure*. That is to say, the F-measure of the contour precision is based on the precision and recall of the contour. This indicator is used to measure the precision of the segmentation boundary or how well the segment contours $c(\hat{Y})$ match the ground-truth contours $c(Y)$. Usually, the value of P_c and R_c can be computed via bipartite graph matching [6], then the boundary accuracy F can be computed as:

$$\mathcal{F} = \frac{2P_c R_c}{P_c + R_c} \tag{7}$$

- **Temporal stability** $\mathcal{T}$: This is informative of the stability of segments. It is computed as the pixel-level cost of matching two successive segmentation boundaries. The match is achieved by minimizing the shape context descriptor [1] distances between matched points while preserving the order in which the points are present in the boundary polygon. Note that $\mathcal{T}$ will compensate motion and small deformations, but not penalize the inaccuracies of the contours [11].
- **Average of Jaccard and Contour Precision** $\mathcal{J}\&\mathcal{F}$:

$$\mathcal{J}\&\mathcal{F} = \frac{\mathcal{J} + \mathcal{F}}{2} \tag{8}$$

4.2. Performance Comparison

Method	$\mathcal{J}$			$\mathcal{F}$			$\mathcal{J}\&\mathcal{F}$
	mean↑	*recall*↑	*decay*↓	*mean*↑	*recall*↑	*decay*↓	*mean*↓
MATNet [18] (UVOS)	56.7	65.2	−3.6	60.4	68.2	1.8	58.6
RTNet [13] (UVOS)	61.6	72.2	-	66.7	75.8	-	-
TransVOS [7] (SVOS)	75.7	-	-	80.5	-	-	78.1

Table 1: Quantitative object-level UVOS and SVOS results on DAVIS_{17} val in terms of region similarity $\mathcal{J}$, boundary accuracy $\mathcal{F}$, and Average of Jaccard and Contour Precision $\mathcal{J}\&\mathcal{F}$

5. Challenges faced throughout the project

Many issues in video object segmentation and tracking are very challenging. In general, VOS has some common challenges, such as background clutter, property changes, the conflict between similar instances, temporal consistency, and the balance between efficiency and accuracy, low resolution, occlusion, deformation, motion blur, scale variation, and more. But there are some specific characteristics determined by the objectives and tasks; for example, objects in the VOS can be complex due to fast motion, out-of-view, and real-time processing. In addition, the effects of heterogeneous object, interacting object, edge ambiguity, and shape complexity must be segmented.

These challenges motivate most current methods and are exploited to improve upon the existing approaches. Below are some common challenges faced during carrying out our experimentation.

- **Object property change**: This challenge mainly affects the VOS methods based on visual similarities. During inference, these methods segment the regions with similar visual features to the target objects annotated (most are SVOS methods) or predicted (most are UVOS methods) in first frame.
- **Occluded by distractors**: Mainly affects the propagation-based VOS methods, which consider the objects predicted in the previous frame to estimate current frame segmentation.
- **Distraction from similar objects/backgrounds**: Affects mainly the visual similarities, saliency or motion patterns. During inference, these methods segment the objects with specific features, e.g., similar visual features to the target objects (most are SVOS methods) or prominent saliency/motion patterns (most are UVOS methods).
- **Temporally consistent VOS**: Affects methods using less motion information. During inference, these methods essentially perform image segmentation for each video frame. Therefore, it is difficult to maintain the temporally consistent of the segmented objects.

6. Conclusion

In this study, we provided a comprehensive overview of the video object segmentation (VOS) through several experimentation ranging from traditional to deep-learning based SoTA methods. We described challenges and potential application in the field, classified and analyzed the recent methods, and discussed different algorithms. We provided a hierarchical categorization of the different groups in existing works and summarized some object representation, image features, motion cues, and more. We also described various pre-process and post-process CNN-based methods and discussed the advantages or disadvantages of the methods. Moreover, we described the related video datasets for VOS and the evaluation metrics of pixel-wise mask and bounding box–based techniques. Finally, the analysis and prospect of each task of VOS was presented. In the future, we will further extend it to other video analysis tasks, such as action recognition and video classification.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[2] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *Proceedings of the 11th European Conference on Computer Vision: Part V*, ECCV'10, page 282–295, Berlin, Heidelberg, 2010. Springer-Verlag.

[3] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixe, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[4] Sergi Caelles, Alberto Montes, Kevis-Kokitsi Maninis, Yuhua Chen, Luc Van Gool, Federico Perazzi, and Jordi Pont-Tuset. The 2018 DAVIS challenge on video object segmentation. *CoRR*, abs/1803.00557, 2018.

[5] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. 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 (CVPR)*, June 2019.

[6] D.R. Martin, C.C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.

[7] Jianbiao Mei, Mengmeng Wang, Yeneng Lin, and Yong Liu. Transvos: Video object segmentation with transformers. *arXiv preprint arXiv:2106.00588*, 2021.

[8] Fang Zheng Mingqi Gao, Caifeng Shan James J. Q. Yu, and Jungong Han Guiguang Ding. Deep learning for video object segmentation: a review. 2022.

[9] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187–1200, 2014.

[10] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 724–732, 2016.

[11] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 724–732, 2016.

[12] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbelaez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *CoRR*, abs/1704.00675, 2017.

[13] Sucheng Ren, Wenxi Liu, Yongtuo Liu, Haoxin Chen, Guoqiang Han, and Shengfeng He. Reciprocal transformations for unsupervised video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15455–15464, June 2021.

[14] Wenguan Wang, Xiankai Lu, Jianbing Shen, David J. Crandall, and Ling Shao. Zero-shot video object segmentation via attentive graph neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[15] Wenguan Wang, Tianfei Zhou, Fatih Murat Porikli, David J. Crandall, and Luc Van Gool. A survey on deep learning technique for video segmentation. *ArXiv*, abs/2107.01153, 2021.

[16] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In Vittorio Ferrari, Cristian Sminchisescu, Martial Hebert, and Yair Weiss, editors, *Computer Vision – ECCV 2018 - 15th European Conference, 2018, Proceedings*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 603–619, Germany, 2018. Springer.

[17] Rui Yao, Guosheng Lin, Shixiong Xia, Jiaqi Zhao, and Yong Zhou. Video object segmentation and tracking: A survey. 11(4), may 2020.

[18] Tianfei Zhou, Jianwu Li, Shunzhou Wang, Ran Tao, and Jianbing Shen. Matnet: Motion-attentive transition network for zero-shot video object segmentation. *IEEE Transactions on Image Processing*, 29:8326–8338, 2020.