Synthesis Lectures on Computer Vision

SYNTHESIS
COLLECTION OF TECHNOLOGY

Ning Xu · Weiyao Lin ·
Xiankai Lu · Yunchao Wei

Video Object Segmentation

Tasks, Datasets, and Methods

Springer

Synthesis Lectures on Computer Vision

This series publishes on topics pertaining to computer vision and pattern recognition. The scope follows the purview of premier computer science conferences, and includes the science of scene reconstruction, event detection, video tracking, object recognition, 3D pose estimation, learning, indexing, motion estimation, and image restoration. As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, or multi-dimensional data from a medical scanner. As a technological discipline, computer vision seeks to apply its theories and models for the construction of computer vision systems, such as those in self-driving cars/navigation systems, medical image analysis, and industrial robots.

Ning Xu · Weiyao Lin · Xiankai Lu ·
Yunchao Wei

Video Object Segmentation

Tasks, Datasets, and Methods

Springer

Ning Xu
Adobe Research
San Jose, CA, USA

Xiankai Lu
School of Software
Shandong University
Jinan City, China

Weiyao Lin
Department of Electronic Engineering
Shanghai Jiao Tong University
Shanghai, China

Yunchao Wei
School of Computer and Information
Technology
Beijing Jiaotong University
Beijing, China

ISSN 2153-1056 ISSN 2153-1064 (electronic)
Synthesis Lectures on Computer Vision
ISBN 978-3-031-44655-9 ISBN 978-3-031-44656-6 (eBook)
https://doi.org/10.1007/978-3-031-44656-6

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Paper in this product is recyclable.

Preface

With the universal access of cameras in all kinds of devices (e.g., mobile phones, surveillance cameras, in-vehicle cameras), video data has gone through exponential increase nowadays. For a lot of video-related applications such as autonomous driving, video editing and augmented reality, segment target objects is important to help understand video content.

Video object segmentation (VOS) is a fundamental task for video understanding in computer vision. VOS can be divided into different settings further, which are slightly different in the definition and focus areas. In this book, we will give a thorough introduction of the task of VOS (including different settings). In Chap. 1, we first briefly introduce the settings of VOS. Then in Chap. 2, we introduce the VOS task under most popular problem settings including semi-supervised VOS, unsupervised VOS, interactive VOS,video instance segmentation, video semantic segmentation and video panoptic segmentation. Finally, in Chap. 3, we give introduction to our held LSVOS challenge on VOS.

San Jose, USA
September 2022

Ning Xu

Contents

Introduction

1

With universal access to cameras in all kinds of devices (e.g., mobile phones, surveillance cameras, in-vehicle cameras), video data has gone through an exponential increase nowadays. The ability to understand, track, and segment objects in videos has become a fundamental and essential problem in various video-related applications. For example, in the field of video and movie editing, it is a very common task to separate the pixels of a foreground apart from the pixels of the background in the original video, and then the foreground pixels are put onto some new background to create fancy visual effects. Moreover, in the field of augmented reality (AR), to attach AR effects to some moving object in the space and to make the effect look realistic, we need some algorithms to follow the motion of the target object in real-time.

The technology behind these applications is called video object segmentation, which is a fundamental problem for video understanding in computer vision. In a formal definition, the task of video object segmentation (VOS) aims at dividing pixels of a video into disjoint subsets where each subset usually represents either a target object or the background. Compared to the traditional video object tracking task, VOS provides finer-grained outputs and focuses more on the object segmentation quality. Moreover, VOS usually works on short video clips, possibly with some amount of manual interaction to guarantee the segmentation output is accurate enough.

In this book, we will give a comprehensive introduction of the VOS task. Specifically, in Chap. 2, we introduce the VOS task under most popular problem settings including semi-supervised VOS, unsupervised VOS, interactive VOS, video instance segmentation, video semantic segmentation, and video panoptic segmentation. For each problem setting in VOS, we begin by providing a formal problem definition. Subsequently, we discuss the prominent challenges, along with the most widely used datasets and evaluation metrics. Then, we provide an overview of the diverse range of methods proposed to tackle the problem.

N. Xu et al., *Video Object Segmentation*, Synthesis Lectures on Computer Vision,
https://doi.org/10.1007/978-3-031-44656-6_1

Finally, we carefully select two or three most representative and effective methods and delve deep into their underlying ideas, detailing their experimental findings. Additionally, in Chap. 3, we present the results of the recent influential LSVOS challenge on VOS, which allows readers to access the methods and outcomes achieved by the top-performing teams, providing valuable insights into the advancements made in the field.

Before delving into the specifics of each task in the subsequent chapters, we provide a comprehensive overview of each task. This enables readers to grasp the objectives and distinctions associated with each task.

- *Semi-supervised VOS (SVOS)* is a VOS setting where a full segmentation mask of the foreground (FG) object is only given in the initial frame, and algorithms need to separate the foreground (FG) and background (BG) masks in all the remaining frames.
- *Unsupervised VOS (UVOS)* is a VOS setting where the manual annotations of the foreground (FG) object are NOT given in any frame, and algorithms need to separate the FG and BG masks in all the frames automatically.
- *Interactive VOS (IVOS)* aims to provide a more flexible setting than SVOS where users can provide various types of FG and BG annotation (*i.e.,* points, scribbles, boxes) to select the objects of interest and gradually refine the segmentation results by providing more interactions.
- *Referring VOS (SVOS)* aims to perform VOS referred by a given language expression describing the FG object instead of a full mask annotation as in SVOS. It thus requires multi-modal understanding between vision and language.
- *Video instance segmentation (VIS)* aims at simultaneous detection, segmentation and tracking of object instances belonging to certain categories in videos. It is different from VOS in that it not only requires object classification (i.e., recognize object categories) but also requires comprehensive labeling of all instances in the video.
- *Video semantic segmentation (VSS)* is a video pixel-level scene parsing task which requires the assigning a class label to every pixel in all frames of a video sequence.
- *Video panoptic segmentation (VPS)* is similar to VSS in that it also requires unique and consistent semantic scene parsing within a video. But it also asks to associate instance IDs for the same objects across frames.

Besides the above differences, we list other attributes that can distinguish these tasks in Table 1.1. First, each task involves a different combination of sub-problems. For example, some VOS tasks do not need to solve the classification problem (i.e., recognize object categories) while VIS, VSS, and VPS need. Second, most tasks need to handle multiple instances in the video. In contrast, VSS does not need to distinguish different instance IDs

Table 1.1 Comparison between VOS tasks on several attributes

Task	Classification	Tracking	Detection	Segmentation	Object num	Background
VOS						
SVOS	×	√	×	√	Multiple	×
UVOS	×	√	√	√	Multiple	×
IVOS	×	√	×	√	Multiple	×
RVOS	×	√	√	√	Multiple	×
VIS	√	√	√	√	Multiple	×
VSS	√	×	×	√	–	√
VPS	√	√	√	√	Multiple	√

and SOT only focuses on a single target object. Lastly, the two scene parsing tasks VSS and VPS need not only semantic labeling of FG objects but also BG while the other tasks do not need semantic understanding of BG.

VOS

2

In this chapter, we elaborate on the task of video object segmentation (VOS), which aims at dividing pixels of a video into disjoint subsets where each subset usually represents either a target object or the background. The VOS task has different problem settings given different input or output requirements. For example, based on the amount of manual annotation used for input, the task can be categorized as semi-supervised VOS Sect. 2.1 where a first-frame annotation for a target object is provided, unsupervised VOS Sect. 2.2 where no target object is indicated and interactive VOS Sect. 2.3 where annotations of the target object can be provided at multiple frames or multiple rounds. While based on the output formats, the task can be categorized as video instance segmentation Sect. 2.4 where the segmentation outputs include all object instances of a predefined list of categories, video semantic segmentation where the outputs do not differentiate object instances within the same category and panoptic segmentation Sect. 2.5 where the outputs also include stuff-like categories such as sky and river.

2.1 Semi-supervised Video Object Segmentation

2.1.1 Introduction

In this section, we introduce a most popular setting of the video object segmentation task, called Semi-supervised Video Object Segmentation (SVOS). The term "semi-supervised" should not be confused with "semi-supervised learning" which usually represents one area of machine learning paradigms which leverage both labeled and unlabeled data in training, as compared to other paradigms such as supervised learning and unsupervised learning. Instead, in our task the term "semi-supervised" indicates that the manual annotation of the foreground (FG) object at the first frame is given as evidence, and algorithms need to rely on it to separate the FG and BG masks in all the remaining frames. It is used to distinguish from

N. Xu et al., *Video Object Segmentation*, Synthesis Lectures on Computer Vision,
https://doi.org/10.1007/978-3-031-44656-6_2

other VOS settings that use different amount of manual annotation such as unsupervised VOS which identify FG automatically without any manual annotation and interactive VOS where manual annotation is provided in an interactive fashion. It is worthy noting that in more recent papers [211] this task is also called "semi-automatic video object segmentation'.

There are several reasons for the recent popularity of the SVOS setting. First, this setting is easy and flexible for many real-world applications. For example, rotoscoping, which aims to separate FG from BG in a movie or video, is an essential but a very tedious step for the video editing industry. Although people invented green/blue screen to make the problem easier, it still requires careful manual annotation on almost every single frame, which could take hours or even days. In contrast, in the SVOS setting, users only need to annotate the FG mask at the very first frame, which saves a lot of manual effort. Even for the first-frame annotation there are many existing image-based interactive segmentation methods [227, 228] to help ease and facilitate this step, which further make this setting more efficient.

Second, SVOS has many connections to another important video understanding problems video object tracking, which will be detailed later in our book. In fact, SVOS and video object tracking share many common problems and challenges and thus it is not surprising that their methods resemble to each other. The improvement and novel ideas of one task can also usually be applied to another, making them inseparable. In addition, SVOS methods are commonly applied to other video applications. For instance, SVOS methods can provide intermediate results to video summarization which leverage visual objects across multiple videos [36] and provide visualization tool to assist video retrieval [168]. In the field of video compression, SVOS methods are used in video-coding standards to implement content-based features and high coding efficiency [92].

In the rest of this section, we will first introduce the common challenges of this task Sect. 2.1.2, popular benchmarks and evaluation metrics Sect. 2.1.3 as well as overview of common and effective methods Sect. 2.1.4. Then we presents three novel SVOS methods in detail which belong to image matching methods Sect. 2.1.5, long-range temporal methods Sect. 2.1.6 and memory methods Sect. 2.1.7 respectively.

2.1.2 Challenges

In this section we discuss a few challenging problems in SVOS.

- *Appearance changes* happen commonly when FG object undergoes large lighting-condition change or reflection. This will pose a challenge to SVOS methods which solely rely on matching object appearance between frames. While methods exploiting frame-dependent motion and connectivity are more robust to this issue.
- *Background changes* usually exist in dynamic moving scenes, *e.g.* running car and camera motion. This will cause difficulty for methods which adopt static-background assumption.

- *Fast motion* is a common problem for methods exploiting motion cues in their frameworks. Many methods use optical flows as conditions to predict object locations while it is well known that optical flows are inaccurate under fast motion. Potential solutions to this issue include either leveraging object appearance similarity to enlarge the search area or learn more reliable motion cues on the fast motion videos.
- *Occlusion* is one of the long-lasting problems for video segmentation. It can cause both drastic appearance change and unreliable motion estimation. Using mask propagation history information is one of the effective ways to alleviate this issue.

From the above discussion, we can see that to have a robust SVOS method, we have to comprehensively leverage appearance similarity, short-range and long-range temporal information as well as mask propagation information. We will give more detailed explanation in the method sections.

2.1.3 Datasets and Metrics

This section will talk about popular public dataset as well as common evaluation metrics.

2.1.3.1 Evaluation Metrics

The two evaluation metrics that are commonly used in SVOS are region similarity $\mathcal{J}$ and the contour accuracy $\mathcal{F}$ [157].

region similarity $\mathcal{J}$. Given an output segmentation M and the corresponding ground-truth mask G.region similarity $\mathcal{J}$ is defined as the $intersection - over - union$ of G and M. $\mathcal{J} = \frac{|G \cap M|}{|G \cup M|}$.

contour accuracy $\mathcal{F}$. As one can interpret a segmentation M as a set of closed contours $c(M)$ delimiting the spatial extent of the mask. Given the contours $c(M)$ and $c(G)$ for two segmentations M and G respectively. One can make a bipartite graph matching between $c(M)$ and $c(G)$ with robustness to small inaccuracies, as proposed in [136]. Based on bipartite graph matching, one can compute the contour-based precision and recall P_c and R_c. The contour accuracy $\mathcal{F}$ is calculated by F1-score. $\mathcal{F} = \frac{2P_c R_c}{P_c + R_c}$. For efficiency, the bipartite matching is approximated via morphology operators.

Compared to $\mathcal{J}$, $\mathcal{F}$ are more sensitive to the contour. If a closed object is segmented to several isolated part, it may get a decent $\mathcal{F}$, but the $\mathcal{J}$ must be low.

Xu et al. [230] further proposed to calculate $\mathcal{J}$ and $\mathcal{F}$ for seen and unseen objects separately to get $\mathcal{J}_{seen}$, $\mathcal{J}_{unseen}$, $\mathcal{F}_{seen}$ and $\mathcal{F}_{unseen}$. Where 'seen' refer to the categories of objects which appear both in training and testing dataset and 'unseen' refer to the categories of objects that only appear in testing dataset. This helps to evaluate the generalization of methods on new objects

Table 2.1 Comparison among existing datasets. "Annotations" denotes the total number of object annotations. "Duration" denotes the total duration (in minutes) of the annotated videos

Scale	JC [51]	ST [104]	YTO [81]	FBMS [150]	DAVIS [157, 159]		YouTube-VOS [229]
Videos	22	14	96	59	50	150	3,252
Categories	14	11	10	16	–	–	78
Objects	22	24	96	139	50	376	6,048
Annotations	6,331	1,475	1,692	1,465	3,440	26242	133,886
Duration	3.52	0.59	9.01	7.70	2.88	8.72	217.21

Beside these two metrics. Perazzi et al. [157] also proposed another metric-Temporal stability $\mathcal{T}$ to evaluate the temporal consistency of predicted segmentation of different frames. In detail, one can transform mask M_t of frame t into polygons representing its contours $P(M_t)$. Then describe each point $p_t^i \in P(M_t)$ using the Shape Context Descriptor (SCD).Next, we pose the matching as a Dynamic Time Warping (DTW) [164] problem, where we look for the matching between p_t^i and p_{t+1}^i that minimizes the SCD distances between the matched points while preserving the order in which the points are present in the shapes.The resulting mean cost per matched point is used as the measure of temporal stability $\mathcal{T}$. However, this metric is not widely used for method comparison (Table 2.1).

2.1.3.2 Public Dataset and Challenge

First, we list and make comparison for all the existing SVOS datasets. Among them, DAVIS [157, 159] and YouTube-VOS [229] become the mainstream benchmark datasets in recent years.

DAVIS. The are two version for DAVIS, DAVIS16 [157] and DAVIS17davis2017. DAVIS2016 is a single-object dataset in which only one object is annotated for a video. It consists of 50 high quality, Full HD video sequences, spanning multiple occurrences of common video object segmentation challenges such as occlusions, motion blur and appearance changes. The total 50 videos are subdivided into a training-set (30 videos) and a test-set (20 videos) for evaluation.

DAVIS17 is multi-object dataset in which more than one objects may be annotated in a video. Some videos are from DAVIS16 while just provided with multi-object annotation. It contain 150 videos totally, and they are subdivided into 4 subset-train, val, test-dev and test-challenge, containing 60, 30, 30 and 30 videos respectively. The annotation of train and val set is available for training and testing individually. The test-dev set is used for long-term evaluation while test-challenge set is used for a timed challenge. DAVIS17 is the first dataset with public challenge held.

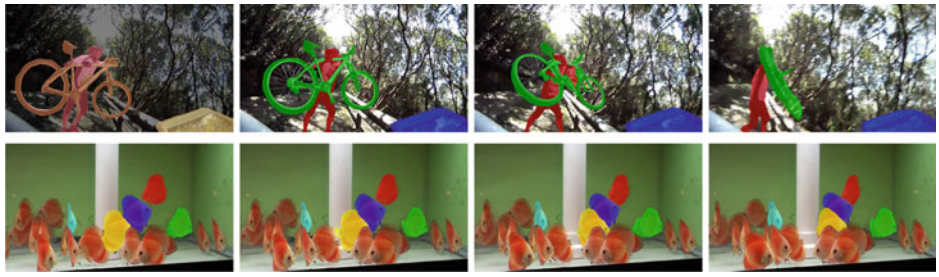

Fig. 2.1 The ground truth annotations of sample video clips in YouTube-VOS. Different objects are highlighted with different colors

YouTube-VOS. YouTube-VOS is the only large-scale SVOS dataset. It contains 3,252 YouTube video clips featuring 78 categories covering common animals, vehicles, accessories and human activities. Each video clip is about 3–6 s long and often contains multiple objects, which are manually segmented by professional annotators. Compared to other existing datasets, YouTube-VOS contains a lot more videos, object categories, object instances and annotations, and a much longer duration of total annotated videos. Based on it, a Large-Scale Video Object Segmentation challenge (LSVOS) is held since 2018. LSVOS is also the only existing SVOS challenge as DAVIS stopped holding challenge since 2020. The figure below show some video clips with ground truth annotations in YouTube-VOS (Fig. 2.1).

2.1.4 Overview of Methods

Based on whether hand-crafted features are used or not, SVOS can be categorized into non-deep-learning based and deep-learning based methods. Although recent development on SVOS mainly focus on deep-learning based methods, we believe it is still valuable to review those non-deep-learning based methods as their key ideas are still very useful to understand the problem.

2.1.4.1 Non-deep-Learning Based Methods

A large body of this type of methods usually leverage spatial-temporal graphs given the natural of the SVOS problem. Specifically, in a graph $\mathcal{G}$ with a set of vertices $\{d_i\} \in \mathcal{D}$ and edges $\{e_{ij}\} \in \mathcal{E}$, each vertex could represent a pixel, a supervoxel, a patch or an object proposal while each edge could represent the pairwise relation between two given vertices. The goal is to partition the graph into disjoint subgraphs and assign a label l_i (e.g., foreground, background) to each vertex. The label assignment is usually done by minimizing an energy function as follows.

$$E = \sum_{d_i \in \mathcal{D}} U(d_i) + \lambda \sum_{e_{ij} \in \mathcal{E}} w_{ij} \cdot V(d_i, d_j) \tag{2.1}$$

where the first term in the equation expresses how likely a single vertex d_i belongs to the label l_i based on its own characteristics (such as appearance) while the second term represents how likely two vertices d_i and d_j have their labels l_i and l_j given their edge relation (such as spatial-temporal smoothness). Finally w_{ij} and λ are weighting parameters to balance different terms.

Previous methods mainly vary in different ways to construct the graph, i.e., the representation of vertex and spatial-temporal connections. In terms of vertex representation, the common ways are pixel, supervoxel, patch and object proposals. For example, Märki et al. [135] propose to incorporate pixel feature representation in a spatial-temporal bilateral grid, which can approximate long-range connection between pixels while only containing tractable number of variables and graph edges. Jain et al. [81] propose to use supervoxel representation, which is the space-time analog of spatial superpixels. The advantage is that it can provide a bottom-up volumetric segmentation that is more sensitive to long-range object boundaries along the temporal axis. They also propose a new pairwise potential in the energy function to account for the new vertex representation. To make the graph computation more efficient, Perazzi et al. [158] propose to construct the graph over a set of object proposals. Their method first generates a large number of object proposals for each frame which are then pruned to retain the high quality proposals. Finally the label assignment is solved by the maximum a posteriori of a conditional random field.

Another important aspect of graph construction is how the spatial-temporal connection between nodes is determined. Fan et al. [52] builds connection between frames by nearest neighbor fields (NNFs) which are computed by PatchMatch [7]. The NNFs can capture the patch similarity in both color and texture and also capture large displacements and non-rigid motion due to the random search of PatchMatch. Badrinarayanan [4] proposes a temporal tree structure which denotes the undirected acyclic graphical model. The tree is often a forest of sub-trees which links patches in adjacent frames through the video sequence. Some methods [158] build up long-range connections using appearance-based methods.

In addition to the different ways of graph construction, previous methods also differ in how they solve the problem. Some methods [3, 52, 135] employ a locally greedy strategy which only considers two or few consecutive frames at a time, while other methods [158, 193] try to find globally optimal solutions by considering all available frame information. One obvious advantage of the locally greedy strategy is the efficiency. In addition, it is also suitable for applications which only send sequential frame data on-the-fly such as video surveillance. While the advantage of the globally optimal strategy is that it can potentially avoid the limitation of short-range connection and obtain better segmentation results.

2.1.4.2 Deep-Learning Based Methods

Recently, deep learning has been applied extensively on the SVOS task and improved the performance greatly compared to those non-deep-learning-based methods. Based on the temporal information leveraged in the deep learning methods, we can categorize them into *image-based template-matching methods*, *short-range temporal methods*, *long-range temporal methods* and *memory networks*. Next we introduce each of the methods in detail.

Image-Based Template-Matching Networks

These methods do not leverage any temporal information (e.g., motion) and only depend on appearance similarity between the first-frame annotated object and other frames. One popular idea is to leverage transfer learning which adapts a pre-trained image segmentation network to a given test video. For example, [16] trains their SVOS network in two steps. The first step called offline pre-training is to learn an image segmentation network by Fully Convolutional Networks (FCN) on large-scale image segmentation datasets. The second step called online fine-tuning is to fine tune the pre-trained network on the data augmentation of the first-frame object annotation. Later, [201] extends the above work to adapt to large changes in object appearance. Their method updates the image segmentation network online using training samples selected based on network confidence and spatial configuration. Despite the high accuracy achieved by this type of methods, one obvious drawback is their computation complexity since the online learning process is required for every new test video and many iterations of network update, which can become infeasible for real applications. To solve this issue, Yang et al. [234] propose to learn a meta network that can change the parameters of the image segmentation network given a single forward pass.

Another common idea is to treat the SVOS task as a pixel matching problem in the learned feature embedding space. Chen et al. [30] propose to learn a FCN embedding network by a modified triplet loss and the learned network is used to find foreground object by nearest neighbor search. Hu et al. [77] use the first-frame annotation as a template to extract FG and BG features and compare a new frame feature with each of them to obtain a comprehensive segmentation. [178] extracts pixel-level features at different layers of their segmentation network and compare the features to the first-frame object.

Short-Range Temporal Networks

Many methods leverage short-range temporal coherence to propagate the object mask of previous frames to subsequent frames. The short-range temporal information can be obtained either explicitly such as optical flows or implicitly learned by the network.

Optical flows represent the motion of pixels between consecutive frames in a video, and thus is very important for many video understanding tasks. Many SVOS methods incorporate pre-trained optical flows networks [44, 78] as part of their network modules. For example, MoNet [222] uses optical flows to align frame features through bilinear interpolation and proposes a new network layer to separate different motions of FG and BG areas in the optical flow field. PReMVOS [131] proposes a SVOS pipeline consisting of object proposal generation, refinement and merging steps. Optical flows are leveraged in the merging step

by warping a predicted object mask at previous frame to current frame. Bao et al. [6] follow the idea of non-deep-learning-based method to construct graphs over videos. The difference is that they use CNN to learn spatial dependencies (e.g. appearance) for the unary term of the energy function while establish the temporal connections by optical flows for the pairwise term.

Many other methods instead leverage implicitly learned short-term temporal information (usually learned by two consecutive frames). MaskTrack [156] learns a simple ConvNet that comprises of two inputs. One is the current input frame while the other one is the previous estimated object mask. They train their model solely on augmented image segmentation dataset to learn the mask refinement between two near frames. OSMN [234] contains a visual network branch and a spatial branch. The visual branch takes in the current frame data to learn object appearance similarity while the spatial branch takes in a coarse location prior provided by the estimated object mask of previous frame. Oh et al. [151] propose a network structure with two encoders with shared parameters while one encoder takes in the combination of first-frame image and annotation and the other encoder takes in the combination of current-frame image and previous predicted mask. In such way, the mask propagation and object detection are performed simultaneously.

Long-Range Temporal Networks

Although short-range temporal information is able to handle near-frame motion, many challenges in SVOS such as heavy occlusion, large-appearance variation, multiple instances require better understanding of long-range temporal information. More and more recent methods start to pay attention to this point.

DyeNet [110] consists of two major modules. One is called Re-ID module which aims to re-identify occluded objects when they re-appear again. The other one is called Re-MP module which mimics the idea of Recurrent Neural Network (RNN) to propagate object masks. Xu et al. [229] formulate the SVOS problem as a classical sequence-to-sequence learning problem. The input sequence is a video sequence together with an initial first-frame object annotation while the output sequence is the desired object segmentation throughout the video. They propose a Long Short-Term Memory (LSTM) network to learn the long-range temporal information automatically without the help of any existing pre-trained motion network such as FlowNet [44]. In addition to the RNN models, Transformers [196] recently have shown their strong capability to solve sequence-to-sequence problems in Natural Language Processing (NLP) tasks such as machine translation. Some recent SVOS method [46] also introduces to the Transformer framework to attend over a history of multiple frames and learn spatial-temporal correspondence.

Memory Networks

Although RNN models theoretically can capture long-range information, Its sequence-to-sequence formulation is not flexible enough to select the most informative frame information. The current top performing methods usually follow the idea of memory networks, which was first proposed in [152]. The method is called Space-Time Memory (STM) model which