Fast Video Object Segmentation with Temporal Aggregation Network and Dynamic Template Matching

Xuhua Huang1∗ Jiarui Xu1∗ Yu-Wing Tai2 Chi-Keung Tang1
1The Hong Kong University of Science and Technology 2Tencent
xhuangat@ust.hk jxuat@ust.hk yuwingtaitem@tencent.com cktang@cs.ust.hk

Abstract
Significant progress has been made in Video Object Segmentation (VOS), the video object tracking task in its finest level. While the VOS task can be naturally decoupled into image semantic segmentation and video object tracking, significantly much more research effort has been made in segmentation than tracking. In this paper, we introduce “tracking-by-detection” into VOS which can coherently integrates segmentation into tracking, by proposing a new temporal aggregation network and a novel dynamic time-evolving template matching mechanism to achieve significantly improved performance. Notably, our method is entirely online and thus suitable for one-shot learning, and our end-to-end trainable model allows multiple object segmentation in one forward pass. We achieve new state-of-the-art performance on the DAVIS benchmark without complicated bells and whistles in both speed and accuracy, with a speed of 0.14 second per frame and J&F measure of 75.9% respectively. Project page is available at https://xuhuaking.github.io/Fast-VOS-DTTM-TAN/.

1. Introduction
Video object segmentation (VOS) is a fine-grained labeling problem aiming to find pixel-level correspondence across frames in a given video, which has broad range of applications including surveillance, video editing, robotics and autonomous driving. In this work, we focus on the semi-supervised VOS setting in which the ground-truth segmentation masks of the target objects in the first frame are given. The task is then to automatically predict the segmentation masks of the target objects for the rest of the video. With the recent advances in deep learning and the introduction of the DAVIS datasets [44, 45], tremendous progress has been made in tackling this semi-supervised VOS task. Notwithstanding, existing state-of-the-art methods are heavily biased toward semantic segmentation in their design and thus they do not leverage the advantages of excellent tracking solutions. We believe that the VOS task can be naturally decoupled into image semantic segmentation and video object tracking:

In semantic segmentation, most works [8, 9, 10, 11] are mainly based on Fully Convolutional Networks [37]. Many VOS pipelines [29, 4, 49, 1, 48, 52, 41] exploit these architectures to produce a segmentation map. However, semantic segmentation is not instance sensitive. In complex scenarios where many different instances share the same semantics (e.g., pedestrians, vehicles and cases in Figure 1), these methods based on semantic segmentation may fail to track individual objects consistently. This has prompted us

∗Equal contribution. This research is supported in part by Tencent and the Research Grant Council of the Hong Kong SAR under grant no. 1620818.
to revisit the whole pipeline from the perspective of Multiple Object Tracking (MOT).

In MOT, most recent approaches [54, 30, 47, 14, 47, 14, 3, 62, 56, 2] have adopted the popular tracking-by-detection model, where objects are first localized in each frame and then associated across frames. This model benefits from the rapid progress in the field of object detection [19, 46, 24, 33, 35, 16, 6], and has led to several popular benchmarks over the past few years, i.e., MOT15~17 [31, 39]. Such a decoupled pipeline also makes it easy to extend or upgrade the tracking system with latest detection techniques.

Our work is motivated by the recently proposed tracker [2] which consists of further extension of the “tracking-by-detection” model without exploiting tracking annotation. Under the semi-supervised setting, even though the first frame ground-truth is given, tracking annotation is still unavailable. We fine-tune the detector with the first frame ground-truth segmentation mask for domain adaptation, whose iterations are proportional to the length of the sequence, and then perform tracking through the rest of the video. We further extend the tracker with our novel Temporal Aggregation Network and Dynamic Time-evolving Template Matching mechanism, both of which make the tracking pipeline more robust to occlusion and appearance changes.

Many leading methods rely on various overly-engineered design and/or heavy modules to improve their system performance while sacrificing run times. For example, PREMVOS [38], the winner of 2018 DAVIS challenge and current champion on DAVIS benchmark, adopts a total of four neural networks, optical flow and merging mechanism, and as a result it takes around 38 seconds to segment one single frame. DyeNet [32], another leading method in VOS, integrates FlowNet [28] for utilizing optical flow information and Bi-directional Recurrent Neural Network (RNN) for mask propagation, leading to a speed of around 2.4 seconds per frame. Though some of these methods can generate excellent results, the long run times and/or high complexity of their system prevent them from practical deployment or extension with advancement of relevant techniques.

In this paper, we present a simple, fast and high performance new baseline for VOS. Instead of formulating video object segmentation as mainly a segmentation problem, we propose to tackle the problem as a tracking problem with segmentation in fine-grained level. The proposed novel Temporal Aggregation Network and Dynamic Time-evolving Template Matching can be easily integrated into this simple pipeline to boost performance.

Our contributions can be summarized as follows:

- We present a simple, easy-to-extend and end-to-end trainable pipeline for VOS task by introducing the “tracking-by-detection” model into VOS which supports multiple object segmentation in one forward pass. To the best of our knowledge, we are the first to introduce “tracking-by-detection” model into video object segmentation as an online method with strong results and high speed;

- We contribute a novel Temporal Aggregation Network and Dynamic Time-evolving Template Matching mechanism, which are entirely online and naturally fit into one-shot learning to boost performance;

- We achieve a new state-of-the-art performance in both speed and accuracy without complicated bells and whistles on the DAVIS benchmark, with a speed of 0.14 second per frame and $F & F$ mean score of 75.9%.

2. Related Work

Semantic Segmentation Recently, state-of-the-art semantic segmentation frameworks based on the fully convolutional network (FCN) [37] have made remarkable progress. Deeplab [8, 9, 10, 11] and context based methods [59, 50, 20, 60, 27] have further improved the performance over the years. However, benchmarks in semantic segmentation [15, 61] are used to evaluate full image semantic labeling task, while VOS belongs to instance segmentation of one or multiple specific target objects. For semantic segmentation, pixels in the same semantics are labeled with same category with no instance information. As for instance segmentation, the combination of both instance detection and pixel-level segmentation are required. The misalignment of these two tasks indicates that the design and method in semantic segmentation may not directly applicable in video object segmentation. Notwithstanding, with this intrinsic difference, many leading entries [29, 4, 49, 1, 48, 52, 41] in DAVIS benchmark still utilize this pipeline and adopt the architecture of semantic segmentation, which as a result make different instances vulnerable to switch and/or drift as shown in Figure 1.

Tracking-by-Detection Model Recent multiple object tracking (MOT) methods are mostly based on the tracking-by-detection model. As object detection has been well studied on its own, the main focus of MOT is on the data association problem. In MOT Benchmark [31, 39], public detectors are shared among all leading entries [47, 56]. This makes all pipelines designs adopting this model easy to extend with the latest detection approaches [19, 46, 24, 33, 35, 16, 6]. Even though MOT and VOS Benchmark settings differ in many aspects, the same intrinsic key is robust tracking ability. However, surprisingly only a few approaches formulate the VOS problem using this model and they do not come with online setting. While PREMVOS [38] may be regarded as some kind of close proximate to the tracking-by-detection model, its design is very complex and heavy, involving optical flow, multiple networks, and the framework cannot be trained end-to-end. Our method adopts the tracking-by-detection model which explicitly introduces detection and association into video object segmentation.
Compared with PReMVOS, our approach is suitable for online setting, easy to extend and end-to-end trainable.

**Video Object Segmentation** Video Object Segmentation (VOS) aims for joint segmentation and tracking. The recently released DAVIS benchmarks [44, 45, 5] have contributed significantly in pushing the frontier of the relevant state-of-the-arts. However, as noted in [48], many of the methods do not fulfill the design goals of being robust, simple, fast and end-to-end trainable. For instance, as the winning entry of DAVIS 2018 Challenge [5], PReMVOS [38] exploits 4 different neural networks working in tandem, optical flow warping and Markov Random Field (MRF) and iterative inference algorithm, taking more than 38s for processing a given frame. The second runner up CINM [1] uses Markov Random Field (MRF) and iterative inference algorithm. DyeNet [32] incorporates template matching into a re-identification network; however optical flow warping and RNN are used in the mask network, thus making it training complicated and computationally demanding. Recently proposed FEELVOS [48] is an extension of MaskTrack [43] which uses previous frames predictions as input for the current frame, which is prone to accumulation error during tracking. Moreover, due to the absence of shared design model, many methods have limited practical usage and are not easy to extend. Our introduction and extension of “tracking-by-detection” model aims to establish a new, simple and strong baseline for video object segmentation.

3. Method

Refer to Figure 2. The feature map of the current frame is extracted through our Temporal Aggregation Network (TAN) which coherently combines spatial and temporal information. TAN aggregates neighboring frame spatial features across time to enrich the current frame feature. On top of the aggregated feature map from TAN, a Region Proposal Network (RPN) is applied to extract region proposals which are likely to contain target objects. With region proposals in the current frame, Hungarian matching [40] is performed between template features in the bank and current detected object features. Initially, the template features are just the features of the first frame ground truth objects. After matching, the most confident region proposal for each object will be chosen and passed to the Mask Head for final segmentation generation. At the same time, if large appearance change occurs within the chosen proposals, the new proposals would serve as new templates for future frames.

We first introduce how we incorporate tracking-by-detection model into the video object segmentation task. Then we will present our proposed Temporal Aggregation Network and Dynamic Time-evolving Template Matching.

3.1. Tracking by Detection

The goal of video object segmentation (VOS) is similar to multi-object tracking (MOT), which is to predict the trajectories of multiple objects at a fine-grained level over time. We denote the set of trajectories $T = \{T_i\}_{i=1}^N$, where the trajectory of the $i^{th}$ object can be represented by a series of bounding boxes together with the masks within, denoted by $T_i = \{b_i^t, m_i^t\}_{t=1}^T$, $b_i^t = [x_i^t, y_i^t, w_i^t, h_i^t]$; $x_i^t$ and $y_i^t$ denote the center location of the target $i$ at frame $t$; $w_i^t$ and $h_i^t$ denote respectively the width and height of the target object $i$; $m_i^t$ denotes the corresponding foreground segmentation mask within corresponding bounding box. In MOT usually only $\{b_i^t\}_{t=1}^T$ is needed, while VOS is more fine-grained and should produce $\{m_i^t\}_{t=1}^T$. Bounding box representation is popular in recent detection pipelines [46, 24] while some non-box based methods [12] have also been proposed. We mainly adopt box based detection pipelines in this paper but note that our method is also compatible with non-box based pipelines.

To output segmentation mask, a light-weight Fully Convolutional Network is attached to the detector; foreground/background segmentation is performed on each detected bounding boxes. For simplicity, we ignore the $\{m_i^t\}_{t=1}^T$ which can be easily generated from $\{b_i^t\}_{t=1}^T$. Under semi-supervised setting of DAVIS Benchmark [44, 45, 5], the tracking targets are given in the first frame. We denote tracking targets as $\{t_i\}_{i=1}^N$.

Our method follows the online tracking-by-detection model [54], which first detects multiple objects in each frame and then associates their identities across frames. Following [4, 29] we first train the whole detector offline with the training set in order to adapt the detector to the DAVIS dataset domain. Before inference, one-shot learning is performed with the given ground-truth segmentation for a few iterations to reduce false alarm. Note that one advantage of our training strategy is that no tracking-specific training is required, which implies that our method can be
ready extended to various applications.

During inference, different from [2] and for simplicity we do not exploit bounding box regressor for tracking. Instead, simple Intersection over Union (IoU) metric is adopted. For segmentation based approaches, IoU is hard to be incorporated into the pipeline since there is no explicit bounding box or instance. Our method first detects multiple objects in each frame and then associates their identities across frames. The detected boxes at frame $t$ with confidence score greater than $\sigma_{det}$ are denoted as $\{b_j^t\}_{j=1}^{N_t} = D_t$. Note that $N_t$ is not necessarily equal to $N$ due to false positives and false negatives output of the detector. At $t = 0$, our tracker initializes the tracklet from the first frame bounding boxes $\{t_i\}_{i=1}^{N} = \{b_j^0\}_{j=1}^{N_0} = D_0$. At frame $t$, a bipartite graph is constructed based on location similarity IoU, the weight $\omega_{loc}$ between $i$th target and $j$th detected object in the current frame is defined as the IoU of $t_i$ and $b_j^t$. After Hungarian matching [40] is performed, the target $t_i$ will be updated with $b_k^t$ and added to $T_j$ if matched. When running the assignment process in a frame-by-frame manner the object trajectories are produced.

This naive tracking-by-detection pipeline under the setting of semi-supervised video object segmentation will be extended in following sections.

3.2. Temporal Aggregation Network

Figure 3 illustrates the Temporal Aggregation Network (TAN) with details to be described in the following.

Many methods in object tracking have exploited temporal information to facilitate tracking. However, most of them require optical flows to obtain pixel-wise alignment [64, 63] which can be quite computationally expensive. Inspired by the recent progress in large-scale action recognition benchmarks [7, 22], we propose a novel Temporal Aggregation Network to incorporate backbones in image classification and video recognition, which align well with the tracking-by-detection model as well as detection in image and tracking in video.

Image Backbone Most object detection methods have exploited ImageNet [17] pretrained backbone for faster convergence as studied in [23]. We use ResNet [25] as the standard setting similarly done in other detection approaches. We denote the outputs of stage 3, 4, 5 as $c_3$, $c_4$, $c_5$ respectively. The input of Image Backbone is the key frame where objects are to be detected.

Video Backbone Inspired by recent progress in Human Action Recognition [7, 51, 18], we believe 3D convolution is effective in utilizing temporal information across consecutive frames, although no previous works have attempted to utilize this technique to tackle VOS task. Note that while optical flow approaches in [38, 32] utilize explicit correspondence, 3D convolution network can directly learn temporal patterns from an RGB stream. According to [7], I3D performs best against other methods such as CNN+LSTM counterpart. Therefore, we adopt I3D in our video backbone and at the same time, similar to [51], I3D$_{3\times1\times1}$ and I3D$_{1\times3\times3}$ are used in order to reduce the computation cost of 3D convolution. We denote the outputs of stage 3, 4, 5 as $i_3$, $i_4$, $i_5$ respectively. The previous $\alpha$ frames and current key frame are concatenated along the temporal axis resulting in a 3D tensor of size $(\alpha + 1) \times H \times W$ which is then fed into the video backbone.

Temporal Aggregation Fusion Inspired by Slow-Fast [18], the separation and fusion of fast/slow video stream is a good practice to make use of temporal information while preserving key frame feature. As conv-add style in [18], the new feature map $f_i$ is computed by $c_i + \mathcal{N}(i_i)$ for stages 3, 4, 5, where $\mathcal{N}$ denotes a small CNN to fuse features with Max Pooling to compress information along the temporal dimension so that the two branches have the same size at each stage. $f_i$ will be used in the subsequent detection networks. Unlike RNN based methods [32, 47] which require sophisticated training schedule, our network is feed-forward and can be jointly trained. As shown in Figure 5 in the experimental section, by incorporating temporal information, our system can better handle occlusion.

3.3. Dynamic Time-evolving Template Matching

Obviously the naive IoU based tracking algorithm is not robust to large camera movement or object deformation. As pointed out in [2, 38], re-identification is also essential to object tracking. Thus we propose the novel Dynamic Time-evolving Template Matching (DTTM) to tackle the re-identification problem. Figure 4 summarizes the method.

Instead of cropping bounding box regions from RGB image space, we prefer the feature produced by the backbone since it is encoded with high-level semantic meaning. Let $A(b^t)$ denote the appearance feature in bounding box $b$ extracted from the backbone feature map of frame $t$ which is a
high dimensional vector. The appearance similarity weight term is defined as

$$\omega_{ij}^{\text{app}} = \frac{A(t_i) \cdot A(b_j^t)}{||A(t_i)|| \cdot ||A(b_j^t)||}$$ \hfill (1)

As pointed out in [47, 56], location cue and appearance cue are essential features in multiple object tracking. Therefore the bipartite graph is constructed using $\omega_{ij} = \omega_{ij}^{\text{loc}} + \omega_{ij}^{\text{app}}$ in order to take both cues into account, where $\omega_{ij}^{\text{loc}}$ has been defined in Section 3.1.

In ideal scenarios, the learned appearance feature from the first frame is sufficient for tracking the rest. However, a constant template clearly does not work in practice especially in long-term tracking, since the target objects may undertake many appearance changes due to deformation, occlusion, etc. So a proper design for time-evolving template is essential, which leads to our online updating template appearance features during the tracking progress.

Moving average is one of the widely adopted approaches for template update across temporal axis. The single hyper parameter momentum controls the proportion of features to be updated or preserved, which is extremely sensitive to the frame rates of different videos. The static averaging process is also obviously sub-optimal due to accumulation error and feature blurring. We propose to update the template feature more dynamically in a discrete manner.

**DTTM** Full algorithm detailing our matching strategy in pseudocodes can be found in the supplementary material. Initially, the template bank for each target $i$ is constructed as $\text{bank}_i \in \{t_i\}$. The matching result is obtained by performing linear assignment between $t_i \in \bigcup_{j=1}^{N_j} \text{bank}_i$ and $b_j^t \in \{b_j\}_{j=1}^{N_j}$ at each frame $t$ by computing $\omega_{ij}^{\text{loc}} + \omega_{ij}^{\text{app}}$. Given foreground confidence score $\text{conf}(b_j^t)$ greater than threshold $\sigma_{\text{conf}}$, when $t_i$ and $b_j^t$ is a match but their appearances are not similar $\omega_{ij}^{\text{app}} < \sigma_{\text{app}}$, which indicates a drastic appearance change and thus a very likely situation of losing track in the upcoming frames, our DTTM will initialize a new template from the latest matched detection $b_j^t$ for the target object $t_i$ and add it into corresponding feature bank, which results in an extended template bank for the future assignment.

To avoid overflow, the least frequently used template will be removed from the $\text{bank}_i$ when the number of templates is larger than some threshold. Both the initial target feature and updated feature can be considered as templates during later matching process. Our system can thus detect potential accumulation error early and thus prevent its adverse effect by updating with high confidence new template feature. Figure 4 illustrates how DTTM can effectively address the deformation problem in VOS, where not only the latest but also previous templates are considered so that abrupt deformation can be well handled. Note that this matching mechanism is entirely online and time-evolving unlike [38], namely, in our DTTM no future information is required and the template bank keeps evolving over time.

**4. Experiments**

**4.1. Dataset and Evaluation Metrics**

We use DAVIS benchmarks [44, 45, 5] to evaluate our method. These benchmarks are challenging, with large variety in scenes, multiple heterogeneous objects and occlusion, and are widely used in evaluating video object segmentation methods.

**DAVIS 2016** DAVIS 2016 comprises of a total of 50 sequences, 3455 annotated frames, all of which were captured at 24fps and Full HD 1080p spatial resolution. Since computational complexity is a major bottleneck in video processing, each sequence has a short temporal extent (about 2–4 seconds) while including all major challenges typically found in longer video sequences.

**DAVIS 2017** After releasing DAVIS 2016 [44], several strong methods have been proposed making the top performance on DAVIS 2016 saturated. DAVIS 2017 was released which is an extension of the former version and a more challenging benchmark. Overall, the new dataset consists of 150 sequences, totaling 10459 annotated frames and 376 objects. One major difference from DAVIS 2016 is that in DAVIS 2017, multiple object tracking is introduced in video object segmentation. In DAVIS 2016, only single objects are annotated for each frame while in DAVIS 2017, multiple object masks in the same frame are annotated, which makes the task more challenging as complex interaction among multiple target objects can cause occlusion and thus change of topology and appearance. Moreover, as Figure 1 shows, many target objects are in the same category and have very similar appearance. Another important challenge in DAVIS 2017 is that target object category in one video can become background in another video, thus demanding high discriminative ability of the tracker to adapt to different target objects given only the first frame annotation. Note that DAVIS 2017 is also the dataset for DAVIS 2019 Challenge, which we will use in the following unless otherwise specified.
4.2. Implementation Details

We use Faster R-CNN [46] as the detector and ResNet-50 [25] as our default backbone unless otherwise specified. The spatial branch backbone is initialized with weights given by ImageNet [17] classification and the detector is pretrained on COCO [36] as done in [38]. As for the temporal aggregation branch, we use 8-frame input I3D baseline provided by Nonlocal Network [51]. In order to incorporate multiple scales during tracking, feature pyramid network (FPN) [34] is adopted in the backbone to merge high-level semantic information from deeper coarse feature map with shallow fine-grained feature map. Following [34], the RPN anchors span 5 scales (feature is extracted from different levels of FPN outputs) and 3 aspect ratios [0.5, 1.0, 2.0]. Since category classification is unavailable in the DAVIS dataset, the last linear classification layer of R-CNN is replaced by a simple fully connected layer for foreground/background classification. This class agnostic R-CNN is more suitable for semi-supervised setting and more general for real applications. Vanilla Fully Convolution Network [37] is adopted as the segmentation head as default. As in Fast R-CNN [21], an RoI is considered positive if its IoU with the ground-truth box is at least 0.5 and negative otherwise. The segmentation is only performed on positive RoIs during both training and testing.

Training To overcome the domain gap between the pretrained dataset and DAVIS, we first train the entire detection network on DAVIS training set. The input images are resized such that their shorter side is 800p which is similarly done in [46] unless otherwise specified. We train on 8 GPUs with 2 images per GPU (effective mini batch size of 16). All layers of backbone except for c1 and c2 of backbone are jointly fine-tuned with detection annotation. Due to the limited batch size, all batch normalization layers are frozen so that the statistics of mean and variance are unchanged during training. Unlike stage-wise training with respect to RPN in [24], end-to-end training as in [34] is adopted in our implementation which yields better results. All models are trained for 100k iterations using synchronized SGD with a weight decay of 0.0001 and momentum of 0.9. The learning rate is initialized to 0.002, which decays by a factor of 10 after 70k iterations and 90k iterations. Other choices of hyper-parameters also follow the setting in [46].

Inference At test time, we mostly follow the setting of Faster-RCNN in [34]. The box prediction is done on RPN proposals, followed by non-maximal suppression [19]. The segmentation head is then applied to the detected bounding boxes above threshold $\sigma_{\text{conf}}$ only. The instance segmentation results are merged into a single segmentation map by ranking corresponding bounding box confidence scores.

4.3. Ablation Study

The ablation study is performed on the DAVIS 2017 dataset. The outputs are resized to 480p for evaluation.
Table 2. Results of Temporal Aggregation Network (TAN) on ImageNet VID validation set. Duplicate denotes simply stacking the same frame to fit the network input size, while continuous means both current and neighbor frames are fed into TAN.

<table>
<thead>
<tr>
<th>method</th>
<th>optical flow</th>
<th>backbone</th>
<th>input</th>
<th>AP\textsubscript{bbox}</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFF [62]</td>
<td>✓</td>
<td>ResNet-101</td>
<td>Continuous</td>
<td>73.0</td>
</tr>
<tr>
<td>FGFA [63]</td>
<td>✓</td>
<td>ResNet-101</td>
<td>Continuous</td>
<td>76.8</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td>ResNet-50</td>
<td>Duplicate</td>
<td>76.0</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td>ResNet-50</td>
<td>Continuous</td>
<td>78.2 \textpm 2.2</td>
</tr>
</tbody>
</table>

4.3.2 Dynamic Time-evolving Template Matching

Besides no matching and IoU matching, we also compare our dynamic template matching with naive moving average with momentum \( mnt \), in which matching is based on \( A(t_i) = (1 - mnt)A(t_j) + mntA(b_j) \) where \( t_i \) and \( b_j \) is a match. As Table 3 shows, by setting \( \theta_{conf} = 0.5 \) and sliding \( \theta_{app} \), our dynamic template matching outperforms the baseline by a large margin. Note that for the DTTM module, if the threshold becomes too low (e.g. 0.3), it may introduce misleading templates. On the other hand, if the threshold becomes too high (e.g. 0.7), it may filter out useful templates. Either case will lead to performance drop. Note that even though moving average can also improve the performance, it is worse than DTTM in general. We argue that due to the high frame rate of DAVIS dataset, it is hard for naive moving average to model appearance changes which makes it more prone to accumulation error.

Qualitative comparison with leading approaches are provided in Figure 6. Multiple major tracking challenges are presented in this example, such as topology change, occlusion, similar texture pattern and semantic. Embedding and segmentation based approaches like [48, 4, 52] failed to distinguish two dogs and even produce shattered segmentation. Moreover [38] failed to incorporate drastic topology (pose) change of the jumping dog resulting in false tracking, while DTTM can easily delineate targets in similar appearance and keep tracking of rapidly deforming objects.

4.3.3 Segmentation Head

Different off-the-shelf design choices of segmentation head have also been investigated. We deploy two types of skip connection, cascade and join, where cascade sequentially adds the output of each layer to the input of next layer as residue, and join simply aggregates all layer outputs by ad-
dation as final output. Table 4 shows the results of different segmentation heads. Note the simple skip connection has significant improvement over the vanilla FCN counterpart, which indicates that our method can be further benefited from advanced segmentation techniques.

### 4.4. Results on Benchmark

Table 5 tabulates the quantitative results in comparison with leading methods with qualitative results presented in Figure 7. We achieve state-of-the-art single model results in both speed and accuracy on DAVIS Benchmark with stronger backbone ResNeXt-101 [55] and deformable convolution [16], both contribute to further improving overall performance, which again justifies the advantage of our simple and easily extensible pipeline. Despite DAVIS dataset, many leading methods exploit YouTube-VOS [57] as pretraining. In Table 6 we demonstrated that our method achieved higher performance without computational demanding YouTube-VOS pretraining. The speed (t/s) reported in Table 6 considers that there are more than 2 objects on average on DAVIS-2017 dataset. STM [42] needs to handle them individually, while we could detect and track all the targets in one pass. We also reported a reasonable result on YouTube-VOS, 73.5%, $J&F$-Mean without carefully selecting hyperparameter.

### 5. Conclusion

Many leading VOS methods are overly complicated utilizing computationally-heavy modules or highly-engineered pipelines leading to limited practical usage. In this paper, we design a new and strong baseline that simultaneously achieves state-of-the-art speed and accuracy, by integrating the tracking-by-detection model into VOS, since VOS can be naturally decoupled into image semantic segmentation and video object tracking. With this design, our method is easy to extend because ongoing advancement in object tracking can further improve our method in future. With the introduction of multiple object segmentation in the DAVIS 2017 challenge, most leading methods at the time supporting multiple object segmentation required extra modification, whereas our method handles multiple object segmentation in one forward pass. On top of our design, we propose the novel Temporal Aggregation Network (TAN) and Dynamic Time-evolving Template Matching (DTTM), and their effectiveness have been demonstrated experimentally. Without bells and whistles, our method achieves a new state-of-the-art result on DAVIS benchmark for VOS. We hope our fast, practical and easy to extend pipeline will serve as a new baseline for future development targeting at higher efficiency, accuracy and scalability in VOS.
References


