Tracking Instances as Queries: The 2^{nd} Place Solution for Video Instance Segmentation Challenge 2021

Shusheng Yang[1,2]*, Yuxin Fang[1]*, Xinggang Wang[1]†, Yu Li[2], Ying Shan[2], Bin Feng[1], Wenyu Liu[1]

[1]School of EIC, Huazhong University of Science & Technology
[2]Applied Research Center (ARC), Tencent PCG

Abstract

Recently, query based deep networks catch lots of attention owing to their fully end-to-end inference advantages and the competitive results on several fundamental computer vision tasks, such as detection and segmentation. However, how to build a query based video instance segmentation framework with elegant architecture and strong performance remains to be settled. In this report, we describe a unified query based video instance segmentation framework, fully leveraging the one-to-one correspondence between instances and queries. The proposed method obtains 52.3% *mAP on the YouTube-VIS* 2021 *dataset with a single end-to-end model and ranks the* 2^{nd} *place in Track* 2 *of the* 3^{rd} *Large-scale Video Object Segmentation Challenge.*

1. Introduction

Video Instance Segmentation (VIS) [14] is an emerging computer vision task and get rapid development since it was proposed. This task extends the traditional instance segmentation to the temporal domain and requires detecting, classifying, segmenting, and tracking visual instances simultaneously in the given videos. Similar to other video based tasks like VOS (Video Object Segmentation) [9, 17] and VOD (Video Object Detection) [11], video instance segmentation provides a natural understanding of video scenes. Achieving accurate and robust video instance segmentation in real-world scenarios can greatly promote the development of video analysis.

Under the inherent relationship between video instance segmentation and instance segmentation, prevalent video instance segmentation methods [1, 2, 3, 7, 14, 15] prefer utilizing off-the-shelf instance segmentation approaches with various modules for inter-frame feature aggregation and temporal instances association. As a result, modern video instance segmentation methods always follow the one-to-many matching between predictions and ground truth instances, thus the inference process is sensitive to manual-designed post-process operators, far from end-to-end. Nevertheless, to associate instances across video frames, current VIS methods [3, 14] require heuristic association approach and bring lots of artificial hyper-parameters.

To remedy these issues, we propose a query based video instance segmentation method, termed QueryVIS (short for *Tracking Instances as Queries*). The proposed method is built upon the leading query based instance segmentation method QueryInst [6], which detect and segment instances under the guidance of queries. Moreover, an elaborate tracking head is introduced to fully leverage the potential of instance queries for the temporal association. With the one-to-one correspondence between instances and queries, QueryVIS inferences with an end-to-end paradigm, and the well-designed tracking head greatly reduce the number of artificial hyper-parameters.

The proposed QueryVIS is evaluated on the YouTube-VIS Challenge 2021, where it achieves 52.3% mAP on the test benchmark and the 2^{nd} place on the final leaderboard. We also conduct experiments on the standard YouTube-VIS 2019 [14] dataset, on which the proposed QueryVIS outperforms a deal of state-of-the-art methods. With a brief framework and competitive performances, we hope QueryVIS can serve as a strong baseline for future research on video instance segmentation.

2. Method

In this section, we explicate the architecture design of QueryVIS in detail. Fig 1 gives an overall illustration of the proposed methods.

*Equal contributions. This work was done while Shusheng Yang was interning at Applied Research Center (ARC), Tencent PCG.
†Corresponding author, E-mail: xgwang@hust.edu.cn.

2.1. Query Based Instance Segmentation

As aforementioned, QueryVIS is built on the top of QueryInst [6], the well-designed query based instance segmentation framework. The overall object detection and instance segmentation pipelines are summarized as follows.

Object Detection. The object detection pipeline can be formulated as:

$$\begin{aligned} \boldsymbol{x}_t^{\texttt{box}} &\leftarrow \mathcal{P}^{\texttt{box}}\left(\boldsymbol{x}^{\texttt{FPN}}, \boldsymbol{b}_{t-1}\right), \\ \boldsymbol{q}_{t-1}^{*} &\leftarrow \mathrm{MSA}_t\left(\boldsymbol{q}_{t-1}\right), \\ \boldsymbol{x}_t^{\texttt{box}*}, \boldsymbol{q}_t &\leftarrow \mathrm{DynConv}_t^{\texttt{box}}\left(\boldsymbol{x}_t^{\texttt{box}}, \boldsymbol{q}_{t-1}^{*}\right), \\ \boldsymbol{b}_t &\leftarrow \mathcal{B}_t\left(\boldsymbol{x}_t^{\texttt{box}*}\right), \end{aligned} \tag{1}$$

$\boldsymbol{q} \in \mathbf{R}^{N \times d}$ indicates the instance query while N and d denote the total number and dimension of instance query, respectively. For bounding box prediction, at stage t, a pooling operator $\mathcal{P}^{\texttt{box}}$ extracts the current stage bounding box feature $\boldsymbol{x}_t^{\texttt{box}}$ from FPN feature $\boldsymbol{x}^{\texttt{FPN}}$ under the guidance of previous stage bounding box prediction $\boldsymbol{b}_{t-1}$. Meanwhile, a multi-head self-attention module MSA_t is applied to the input query $\boldsymbol{q}_{t-1}$ to get the transformed query $\boldsymbol{q}_{t-1}^{*}$. Then, a box dynamic convolution module $\mathrm{DynConv}_t^{\texttt{box}}$ takes $\boldsymbol{x}_t^{\texttt{box}}$ and $\boldsymbol{q}_{t-1}^{*}$ as inputs and enhances the $\boldsymbol{x}_t^{\texttt{box}}$ by reading $\boldsymbol{q}_{t-1}^{*}$ and generates $\boldsymbol{q}_t$ for the next stage. Finally, the enhanced bounding box feature $\boldsymbol{x}_t^{\texttt{box}*}$ are fed into the box prediction branch $\mathcal{B}_t$ for current stage bounding box prediction $\boldsymbol{b}_t$.

Instance Segmentation. For instance mask prediction, a region-wise pooling operator $\mathcal{P}^{\texttt{mask}}$ extracts the current stage mask feature $\boldsymbol{x}_t^{\texttt{mask}}$ from FPN feature $\boldsymbol{x}^{\texttt{FPN}}$, under the guidance of current stage bounding box prediction $\boldsymbol{x}_t$. A mask dynamic convolution module $\mathrm{DynConv}_t^{\texttt{mask}}$ enhances the original mask feature $\boldsymbol{x}_t^{\texttt{mask}}$ and generates $\boldsymbol{x}_t^{\texttt{mask}*}$. Afterwards, current stage mask head $\mathcal{M}_t$ generates the instance level mask prediction $\boldsymbol{m}_t$ by a stack of convolutional layers. The overall procedure of instance mask generation can be formulated as follows:

$$\begin{aligned} \boldsymbol{x}_t^{\texttt{mask}} &\leftarrow \mathcal{P}^{\texttt{mask}}\left(\boldsymbol{x}^{\texttt{FPN}}, \boldsymbol{b}_t\right), \\ \boldsymbol{x}_t^{\texttt{mask}*} &\leftarrow \mathrm{DynConv}_t^{\texttt{mask}}\left(\boldsymbol{x}_t^{\texttt{mask}}, \boldsymbol{q}_{t-1}^{*}\right), \\ \boldsymbol{m}_t &\leftarrow \mathcal{M}_t\left(\boldsymbol{x}_t^{\texttt{mask}*}\right), \end{aligned} \tag{2}$$

Bipartite Matching. Following [5, 6], we adapt hungarian matching to build the one-to-one correspondences between predictions and ground truth instances. The matching cost of Hungarian matcher is defined as:

$$\mathcal{L}_{Hungarian} = \lambda_{cls} \cdot \mathcal{L}_{cls} + \lambda_{L1} \cdot \mathcal{L}_{L1} + \lambda_{giou} \cdot \mathcal{L}_{giou} \tag{3}$$

where $\mathcal{L}_{cls}$, $\mathcal{L}_{L1}$ and $\mathcal{L}_{giou}$ indicate the focal loss, L1 loss and generalized `IoU` loss, respectively. λ_{cls}, λ_{L1} and λ_{giou} are set as the same as [6].

2.2. Contrastive Tracking Head

Dynamic Instance Embedding. To perform temporal instances association, we first embed all instances to a latent space by a dynamic instance embedding head. Specifically, the embedding process can be formulated as follows:

$$\begin{aligned} \boldsymbol{x}_t^{\texttt{track}} &\leftarrow \mathcal{P}^{\texttt{track}}\left(\boldsymbol{x}^{\texttt{FPN}}, \boldsymbol{b}_t\right), \\ \boldsymbol{x}_t^{\texttt{track}*} &\leftarrow \mathrm{DynConv}_t^{\texttt{track}}\left(\boldsymbol{x}_t^{\texttt{track}}, \boldsymbol{q}_{t-1}^{*}\right), \\ \boldsymbol{e}_t &\leftarrow \mathcal{T}_t\left(\boldsymbol{x}_t^{\texttt{track}*}\right), \end{aligned} \tag{4}$$

Similar to mask prediction, firstly, a region-wise pooling operator extracts instance feature $\boldsymbol{x}_t^{\texttt{track}}$, a track dynamic convolution module $\mathrm{DynConv}_t^{\texttt{track}}$ enhances the instance feature under the guidance of instance query. Then, a linear projection module $\mathcal{T}_t$ projects $\boldsymbol{x}_t^{\texttt{track}*}$ to a latent space and generates instance emebdding $\boldsymbol{e}_t$.

Contrastive Learning. Following [14, 15], we takes a pair of frames as inputs to train the tracking head. During training, the frame pairs are randomly sampled from a training video. One of the frames is picked as key frame, which is fed to the instance segmentation network to get a set of instance predictions. While the other frame is treated as a reference frame, which aims to provide ground truth identities and reference instance embeddings. Assuming there is a detected instances $\mathcal{I}_i$ at the key frame, and there are N already identified instances in the reference frame. It's clear that there is at most one existing identity in reference frame can be assigned to the detected instances. The probability of assigning label n to detected instance $\mathcal{I}_i$ can be formulated as:

$$p_i(n) = \begin{cases} \dfrac{\exp\left(\mathbf{e}_i^{\top}\mathbf{e}_n\right)}{1 + \sum_{j=1}^{N} \exp\left(\mathbf{e}_i^{\top}\mathbf{e}_j\right)} & \text{if } n \in [1, N], \\ \dfrac{1}{1 + \sum_{j=1}^{N} \exp\left(\mathbf{e}_i^{\top}\mathbf{e}_j\right)} & \text{otherwise,} \end{cases} \tag{5}$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ denote the instance embedding of $\mathcal{I}_i$ and n instance embeddings in reference frame. Different from [14], which introduces a cross entropy loss function to optimize the tracking head, QueryVIS adapts a contrastive focal loss to reduce the conflict of multi-task learning. Specifically, the loss function for tracking heads is defined as follow:

$$p_i^{*}(n) = \begin{cases} p_i(n) & \text{if } \mathcal{I}_i = \mathcal{I}_n, \\ 1 - p_i(n) & \text{otherwise,} \end{cases} \tag{6}$$

$$\mathcal{L}_{track} = -\alpha_t (1 - p_i^{*}(n))^{\gamma} \log(p_i^{*}(n)), \tag{7}$$

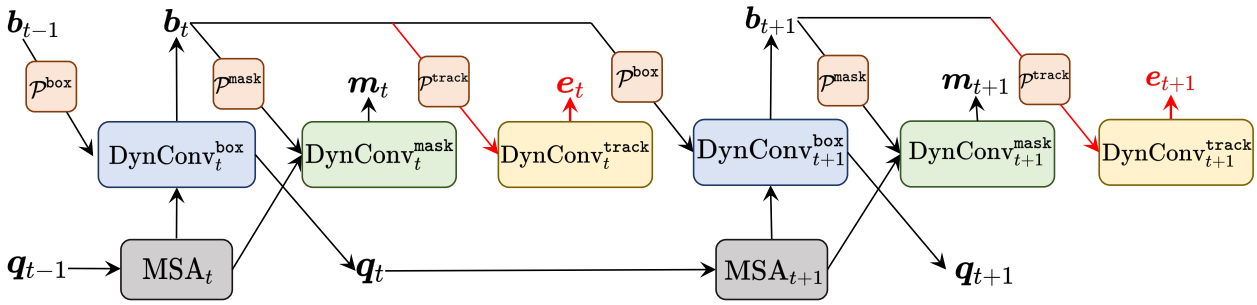

Figure 1. Overall pipeline of QueryVIS. The black arrows indicate the original pipeline of QueryInst [6], while the red arrows stand for the introduced track pipelines to tackle the video instance segmentation problem.

2.3. Online Instance Association

Tracking instances across video frames purely based on the instance embedding is non-trivial as appearance similarity might be confused by instance deformation, occlussion and background change. Similar to [14, 16], QueryVIS leverages several tracking clues such as spatial similarity, detection confidence and category consistency to perform better instance association. Specifically, assume there are M candidate instances and N already identified instances, the matching factor between one candidate instance $m \in [1, M]$ one identified instance $n \in [1, N]$ can be formulated as:

$$\mathcal{F}_{m,n} = \mathcal{S}_{m,n} \cdot \frac{1 + \texttt{IoU}(\texttt{b}_m, \texttt{b}_n)}{2} \cdot \frac{1 + \pi_m}{2} \cdot \delta(\texttt{c}_m, \texttt{c}_n) \quad (8)$$

where $\texttt{IoU}(\texttt{b}_m, \texttt{b}_n)$ indicates the bounding box IoU (intersection over union) between candidate instance m and identified instance n, π_m indicates the detection confidence of candidate instance m, and $\delta(\texttt{c}_m, \texttt{c}_n)$ is an indicator function which gets 1 when the two instances have the same category predictions ($\texttt{c}_m = \texttt{c}_n$) and gets 0 otherwise. $\mathcal{S}_{m,n}$ indicates the normalized appearance similarity between two instances. Specifically, the similarity is normalized by a bi-directional softmax, the computation process can be formulated as follows:

$$\mathcal{S}_{m,n} = \left(\frac{\exp\left(\mathbf{e}_m^\top \mathbf{e}_n\right)}{\sum_{k=1}^{N} \exp\left(\mathbf{e}_m^\top \mathbf{e}_k\right)} + \frac{\exp\left(\mathbf{e}_m^\top \mathbf{e}_n\right)}{\sum_{k=1}^{M} \exp\left(\mathbf{e}_k^\top \mathbf{e}_n\right)} \right) / 2 \quad (9)$$

3. Experiments

3.1. Datasets

We mainly evaluate the proposed QueryVIS on YouTube-VIS 2021 dataset, which is also the standard dataset of YouTube-VIS Challenge 2021. Besides, we also report the system level comparisons between QueryVIS and several state-of-the-art methods on YouTube-VIS 2019 dataset.

3.2. Implementation Details

Training Setup. The basic training setup of QueryVIS is mainly following the original QueryInst [6]. Specifically, the R-CNN head of QueryVIS contains 6 stages and the total number of queries is set to 300. We adapt the recently proposed transformer network [4, 12] as backbone, and use COCO pre-trained weights for parameter initialization. The training process on YouTube-VIS consists of 12 epochs in total. For each iter, the batch size is set to 32 and we use AdamW optimizer with an initial learning rate of 1.25×10^{-5}. The learning rate decreases by 10 at 9^{th} and 11^{th} epoch. Data augmentation includes random flip, multi-scale input, and random crop. Input images are resized such that the shorter side is at least 320 and at most 800, while the other side no longer than 1333.

Inference. Since most of the videos in both YouTube-VIS 2019 [14] and YouTube-VIS 2021 have no more than 10 video instances, during inference we only extract the top 10 instance predictions as valid candidates. The instance masks are generated from the final stage mask head, and the final stage tracking head is used to associate temporal instances. All input images during the inference stage are resized to have their shorter side being 640 and their longer side no longer than 1333.

3.3. Main Results

Tab. 1 shows the results in the final leaderboard of YouTube-VIS Challenge 2021. With a single model, our QueryVIS achieves 52.3 mAP in the test set of YouTube-VIS 2021, and wins the 2^{nd} place in YouTube-VIS Challenge 2021.

Tab. 2 shows the system level comparisons between QueryVIS and state-of-the-art video instance segmentation methods. As shown in the table, QueryVIS outperforms previous state-of-the-art video instance segmentation methods by a large margin.

Team	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
tuantng	54.1	74.2	61.1	43.3	58.9
Ours	**52.3**	**76.7**	**57.7**	**43.9**	**57.0**
vidit98	49.1	68.1	54.5	41.0	55.0
linhj	47.8	69.3	52.7	42.2	59.1
hongsong.wang	47.6	68.4	52.9	41.4	54.6
gb7	47.3	66.5	51.5	40.5	51.6
zfonemore	46.1	64.4	51.0	38.3	50.6
DeepBlueAI	46.0	64.6	52.0	38.7	54.2
zhangxuan	41.0	62.0	42.9	37.3	47.1
Suqi.lmh	32.3	48.8	36.2	30.2	38.2

Table 1. Results in the YouTube-VIS Challenge 2021, compared to top 10 other participants. Our results are highlighted in **bold**.

Method	AP	AP_{50}	AP_{75}
MaskTrack R-CNN [14]	30.3	51.1	32.6
SipMask-VIS [3]	33.7	54.1	35.8
STEm-Seg [1]	34.6	55.8	37.9
CompFeat [7]	35.3	56.0	38.6
CrossVIS [15]	36.6	57.3	39.7
VisTR [13]	40.1	64.0	45.0
IFC [8]	44.6	69.2	49.5
MaskProp [2]	46.6	51.2	–
SeqMask R-CNN [10]	47.6	71.6	51.8
QueryVIS	**52.7**	**78.9**	**57.9**

Table 2. Comparisons with state of the art methods on YouTube-VIS 2019 dataset. Our results are highlighted in **bold**.

4. Conclusion

We report a query based end-to-end framework to tackle the video instance segmentation problem. We build our method upon the state-of-the-art instance segmentation network QueryInst [6] with an elaborate tracking head. Despite the concise framework, the proposed QueryVIS performs strong results and achieves the 2^{nd} place in the YouTube-VIS Challenge 2021. We also conduct experiments on the YouTube-VIS 2019 [14] dataset and find QueryVIS can beat most state-of-the-art VIS methods.

References

[1] Ali Athar, S. Mahadevan, Aljosa Osep, L. Leal-Taixé, and B. Leibe. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *ECCV*, 2020. 1, 4

[2] Gedas Bertasius and Lorenzo Torresani. Classifying, segmenting, and tracking object instances in video with mask propagation. In *CVPR*, 2020. 1, 4

[3] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Sipmask: Spatial information preservation for fast image and video instance segmentation. In *ECCV*, 2020. 1, 4

[4] Jiemin Fang, Lingxi Xie, Xinggang Wang, Xiaopeng Zhang, Wenyu Liu, and Qi Tian. Msg-transformer: Exchanging local spatial information by manipulating messenger tokens. *arXiv:2105.15168*, 2021. 3

[5] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. *arXiv preprint arXiv:2106.00666*, 2021. 2

[6] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. *arXiv preprint arXiv:2105.01928*, 2021. 1, 2, 3, 4

[7] Yang Fu, Linjie Yang, Ding Liu, Thomas S Huang, and Humphrey Shi. Compfeat: Comprehensive feature aggregation for video instance segmentation. *arXiv preprint arXiv:2012.03400*, 2020. 1, 4

[8] Sukjun Hwang, Miran Heo, Seoung Wug Oh, and Seon Joo Kim. Video instance segmentation using inter-frame communication transformers. *arXiv preprint arXiv:2106.03299*, 2021. 4

[9] Yu Li, Zhuoran Shen, and Ying Shan. Fast video object segmentation using the global context module. In *ECCV*, 2020. 1

[10] Huaijia Lin, Ruizheng Wu, Shu Liu, Jiangbo Lu, and Jiaya Jia. Video instance segmentation with a propose-reduce paradigm. *arXiv preprint arXiv:2103.13746*, 2021. 4

[11] Lijian Lin, Haosheng Chen, Honglun Zhang, Jun Liang, Yu Li, Ying Shan, and Hanzi Wang. Dual semantic fusion network for video object detection. In *ACMMM*, 2020. 1

[12] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 3

[13] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. *arXiv preprint arXiv:2011.14503*, 2020. 4

[14] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019. 1, 2, 3, 4

[15] Shusheng Yang, Yuxin Fang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Crossover learning for fast online video instance segmentation. *arXiv preprint arXiv:2104.05970*, 2021. 1, 2, 4

[16] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *arXiv preprint arXiv:2004.01888*, 2020. 3

[17] Qiang Zhou, Zilong Huang, Lichao Huang, Yongchao Gong, Han Shen, Wenyu Liu, and Xinggang Wang. Motion-guided spatial time attention for video object segmentation. In *ICCVW*, 2019. 1