1st Place Solution for YouTubeVOS Challenge 2021: Video Instance Segmentation

Thuy C. Nguyen[1†] Tuan N. Tang[1†] Nam LH. Phan[1] Chuong H. Nguyen[1]
Masayuki Yamazaki[2] Masao Yamanaka[2]
[1]CyberCore AI
[2]Toyota Motor Corporation
{thuy.nguyen, tuan.tang, nam.phan, chuong.nguyen}@cybercore.co.jp, {masayuki.yamazaki, masao.yamanaka}@toyota-tokyo.tech

Abstract

Video Instance Segmentation (VIS) is a multi-task problem performing detection, segmentation, and tracking simultaneously. Extended from image set applications, video data additionally induces the temporal information, which, if handled appropriately, is very useful to identify and predict object motions. In this work, we design a unified model to mutually learn these tasks. Specifically, we propose two modules, named Temporally Correlated Instance Segmentation (TCIS) and Bidirectional Tracking (BiTrack), to take the benefit of the temporal correlation between the object's instance masks across adjacent frames. On the other hand, video data is often redundant due to the frame's overlap. Our analysis shows that this problem is particularly severe for the YoutubeVOS-VIS2021 data. Therefore, we propose a Multi-Source Data (MSD) training mechanism to compensate for the data deficiency. By combining these techniques with a bag of tricks, the network performance is significantly boosted compared to the baseline, and outperforms other methods by a considerable margin on the YoutubeVOS-VIS 2019 and 2021 datasets.

1. Introduction

In this technical report, we present a solution for the task of Video Instance Segmentation (VIS), specifically targeting the VIS dataset hold by the CVPR2021-YoutubeVOS 2021 Workshop. VIS, first introduced in the YoutubeVOS 2019 challenge [19], aims to perform object detection, instance segmentation, and object tracking across video frames. There are 2883 videos with 40 categories in the original 2019 version. In 2021, the dataset is enriched with more than 3800 videos, each has about 30 frames, and the categories are also refined.

[†]equal contribution

VIS by its nature is a multi-task learning problem, and generally there two main approaches. A straightforward way is to perform each individual task separately and sequentially [14]. However, since the components are trained and inferred independently, the full pipeline is complicated, slow, and sub-optimal. The second approach [17, 6, 5] aims to build a single model that jointly learns and performs all the tasks simultaneously. This not only simplifies the pipeline, reduces the inference latency, but potentially improves the final performance.

Our solution also follows the unified direction but is designed to address several specific technical challenges for the dataset. We also hope that it can serve as a strong baseline for more general applications. Our main contributions for the challenge are summarized as follows:

- Our data analysis shows that only a small portion (17%) of the training images are useful, while the rest (83%) are ineffective. We hence propose a training mechanism, named Multi-Source Data (MSD), which could both increase the diversity of data and improves model generalization.
- We exploit multi-task learning and propose the Temporally Correlated Instance Segmentation (TCIS) module to learn the temporal relation between instance masks over adjacent frames.
- We suggest a Bidirectional Tracking (BiTrack) post-processing step to track objects in both forward and backward order to recall more objects, before merging two sets of tracks to obtain a final result.
- Our method secures the 1st rank on the YoutubeVOS-VIS2021, with the score 0.575 mAP on the public validation set, and 0.541 mAP on the private test set. Evaluating the YouTubeVOS-VIS2019 dataset, our solution also obtains 0.543 mAP, setting a new record for the benchmark.

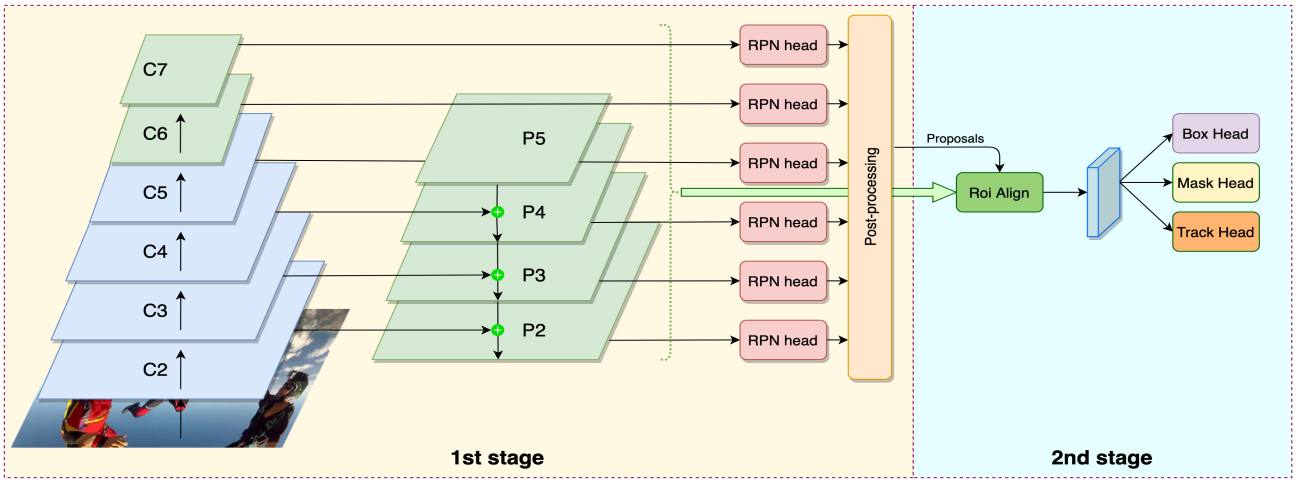


Figure 1. Our baseline framework is based on Mask-RCNN [8]. It has a Backbone, a Feature Pyramid Network with 6 levels, an RPN head in the first stage. The second stage has three branches: object bounding box detection, instance mask segmentation, and object track heads.

The paper is organized as follows. Section 2 summarizes our baseline method, including the model architecture, training, and inference pipeline. Section 3 describes our main solutions, including the data analysis and techniques to improve data diversity, the TCIS component, and a bag of useful tricks to further improve the results. The implementation details, ablation study for different components, and comparison with other methods are presented in section 4. Section 5 concludes our paper.

2. Baseline

Network architecture Our model is built upon Mask-RCNN [8], as illustrated in Fig. 1. The network includes a backbone and a Feature Pyramid Network (FPN) [10] to extract features. A Regional Proposal Network (RPN) is used in the first stage to detect object regions. Given the proposal boxes, the second stage uses a RoI-Align operator to crop features and feed to 3 sub-networks, namely the Box Head for detection, Mask Head for Segmentation, and Track Head [16] to extract embedding vector for object association. Here, we also add an extra level P7 to the FPN, resulting in 6 levels in total.

Training pipeline To train the tracking module, we feed a pair of frames (X_t, X'_t), where X_t is the key frame at time t, and X'_t is randomly sampled within the interval $[t - \Delta, t + \Delta]$. Since they are significantly overlapped, either one of the frames is sufficient for training the detection and the segmentation modules.

Inference pipeline Given a video, the inference is sequentially performed to obtain object attributes (box, label, mask, and embedding). Meanwhile, the data association is conducted online to link the same objects across frames. Finally, we can construct series of unique object masks in the video to output final results.

3. Proposed method

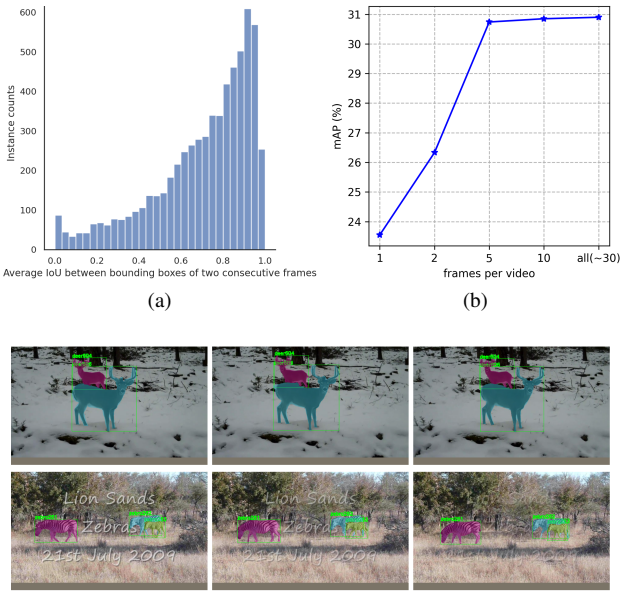
3.1. Data analysis

The YoutubeVOS-VIS2021 dataset has about 90k images, extracted from 3k different videos. However, because the camera may be fixed, and the objects can stay idle or move slowly, the frames in a video can be extremely overlapped, as illustrated in Fig. 2c. Therefore, we conduct two experiments to analyze the data efficiency.

Firstly, we study the severity of overlapping due to object's slow motion and fixed camera. Specifically, we calculate the Intersection over Union (IoU) of each object's bounding box in two consecutive frames, and then take the average IoU score over the video. The IoU histogram of the dataset is then shown in Fig. 2a. We see that the portion of objects having IoU overlap above 0.8 is dominant, verifying that the object displacement is indeed trivial and the overlap is severe. In the second experiment, we uniformly sample different number of frames (eg. 1,2,5,10) in a video to train the model and compare with the results using all frames. For each key-frame, we apply affine transforms to generate a pseudo reference frame for tracking. As shown in Fig. 2b, using only 1 frame in the video already achieves 23.6% mAP, while 5 frames can reach 30.7% mAP, almost equal if using all frames (30.9% mAP). This confirms that 83.3% of the data is redundant and basically ineffectual.

3.2. Multi-Source Data

To enrich the dataset, we utilize a subset from OpenImage [1], that has common object categories with YoutubeVOS-VIS2021, such as bird, fish, turtle. This adds $14k$ images to the dataset. We also combine with the MS COCO 2017 dataset [11], yielding approximately $221k$ images in total. However, using a heterogeneous dataset



(a) (b) (c)

Figure 2. Data efficiency analysis. **(a)** Histogram of bounding boxes's IoU of the same objects appearing in two adjacent-frames. The higher IoU values, the more static objects. **(b)** Accuracy of models trained with different number of frames. **(c)** Many frames in a video are almost identical.

brings some technical issues due to the label difference, that is, no tracking labels in both OpenImage and COCO, low quality or missing ground truth mask in OpenImage, and class mismatch between Youtube VIS and COCO. We address the problems as follows.

Semi-supervised Tracking learning To overcome the absence of ground truth tracking labels, we generate pseudo track-ids by applying augmentations such as shift, rotate, and flip on the key-frame to get its transformed version. Boxes of the same object in different transformed images are assigned with the same and unique track-id.

Weakly-supervised Segmentation learning Images in OpenImage dataset can have no or noisy segmentation mask. Hence, we ignore the segmentation loss of these samples and use them only for detection and tracking training.

Dataset Fusion with Auxiliary Classes The YoutubeVOS-VIS2021 and the MS COCO 2017 datasets have 40 and 80 classes, respectively, and they share 22 categories in common. Conveniently, we can simply ignore the objects of the remaining classes. However, COCO has high-quality labels, especially segmentation masks. Ignoring these classes discards a majority of the dataset while learning all of them will shift our model's target attention. Therefore, to utilize all the available labeled samples, we propose to relax their categories, casting the problem as dataset fusion with auxiliary classes. Following [15], we assume that the remaining 58 classes from COCO can be grouped into K *auxiliary* classes, leading to predict $40 + K$ classes totally. However, we do not manually assign the category for these K classes, but let the network learn the concept of auxiliary classes implicitly and automatically. The proposed method is described in Algorithm 1.

Algorithm 1: Dataset fusion with auxiliary classes

Input: Batch size N; Predicted probs $\hat{y} = \{\hat{y}_i\}_{i=1}^{N}$; Labels $l = \{l_i\}_{i=1}^{N}$;
Output: New labels $y = \{y_i\}_{i=1}^{N}$
for *i in range(N)* **do**
  if $l_i = C + 1$ *(#if it is auxiliary class)* **then**
    if $C < \text{argmax}(\hat{y}_i) \leq C + K$ **then**
      # it is doing correct, continue
      $y_i = \text{argmax}(\hat{y}_i)$
    else
      # randomly pickup among K classes
      $y_i = \text{uniform}(C + 1, C + K)$
    end
  else
    $y_i = l_i$
  end
end

Concretely, in the case of auxiliary classes, the category is selected based on the corresponding prediction. If the predicted index falls into the auxiliary indices, the predicted index is the label, otherwise, the label is randomly sampled in range $[41, 40 + K]$. This mechanism can benefit from two aspects. First, if K is set to 1, the number of samples of this class will be significantly imbalanced with our target classes. In addition, the concept of this class is hard to learn, due to the inconsistency of the feature. Secondly, by randomly sampling class indices, we ensure that the model does not bias to a specific class index, resulting in the extreme case $K = 1$.

3.3. Temporally Correlated Instance Segmentation

Simply applying the techniques from image to video is generally less effective, since the temporal correlation is not taken into account. In fact, we observe that an instance mask in a reference frame is highly related to the corresponding instance mask in a key frame. Consequently, we introduce the module named Temporally Correlated Instance Segmentation (TCIS) to exploit this feature.

Figure 3 depicts the TCIS architecture, in which the Correlation Transform is a standard ResNet block. Let F_i^t and $F_i^{t+\Delta}$ be the RoI feature of the same instance i^{th} at frame t and $t + \Delta$, respectively. F_{mi}^t is the ground truth mask of instance i at frame t. The TCIS module operates by the following steps:

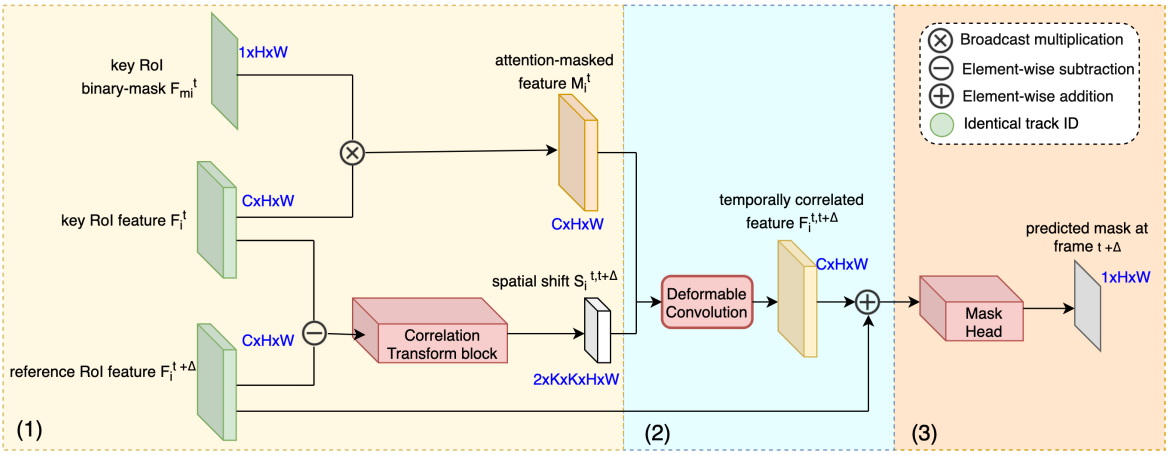


Figure 3. Illustration of TCIS module (best viewed in color). Step 1 computes attention-masked feature M_i^t and spatial shift $S_i^{t,t+\Delta}$ as the feature and deformable-offsets inputs, which are fed to the deformable convolution to compute correlated feature $F_i^{t,t+\Delta}$ in Step 2. Step3 predicts mask for the instance in the reference frame.

1. We multiply the feature F_i^t with its ground truth mask F_{mi}^t to create the attention-masked feature M_i^t. Meanwhile, we subtract F_i^t for $F_i^{t+\Delta}$, and fed the difference to the Correlation Transform block to compute the spatial shift $S_i^{t,t+\Delta}$.

2. Deformable Convolution [4] receives the spatial shift $S_i^{t,t+\Delta}$ as the offset input and the attention-masked M_i^t as the feature input, outputs the temporally correlated feature $F_i^{t,t+\Delta}$. The offset $S_i^{t,t+\Delta}$ helps TCIS pay attention to motion of the instance i between the two frames.

3. Finally, $F_i^{t,t+\Delta}$ and $F_i^{t+\Delta}$ is added together and used as the input for the mask head. We share the same mask head between TCIS and the main branch.

Our TCIS is borrowed from the module MaskProp [2]. The difference is that, objects' features in MaskProp are jointly computed on the whole image, while our approach processes each ROI instance independently. Moreover, TCIS is employed as an auxiliary task during training only, hence induces no additional computation at inference.

3.4. Bidirectional Tracking

Since a motion can happen both forward and backward in time, we propose a post-processing step called Bidirectional Tracking (BiTrack) to further enhance the prediction consistency, as described in Algorithm 2.

Concretely, we first predict objects' bounding boxes, class scores, segmentation masks, and embeddings for all frames in a video. We then run the object ID association backward and forward, matching new objects with existing objects stored in a buffer. If the new object is matched with the existing object, ID of the existing object is assigned to the new object. Otherwise, a new ID will be assigned. The process continues until all frames are checked.

Consequently, we obtain the tracklets F from the forward order, and the tracklets B from the backward order. We consider a frame as valid if it has objects detected in the

Algorithm 2: Bidirectional Tracking

Input:
- Forward tracklets F.
- Backward tracklets B.
- $IsOverlap(f, b)$: the function used to check if tracklet f and tracklet b are overlapped.

Output: Final tracklets M

Initialization

```
    // Init empty matched lists for
       forward tracklets and backward
       tracklets
    F̂ ← ∅, B̂ ← ∅
end
for f in range(F) do
    for b in range(B) do
        if b ∉ B̂ and IsOverlap(f,b) then
            m = Merge(f,b) ;
            M ← M ∪ m ;
            B̂ ← B̂ ∪ b ;
            F̂ ← F̂ ∪ f ;
        else
            continue
        end
    end
end
M ← M ∪ (F \ F̂) ∪ (B \ B̂) ;
return M
```

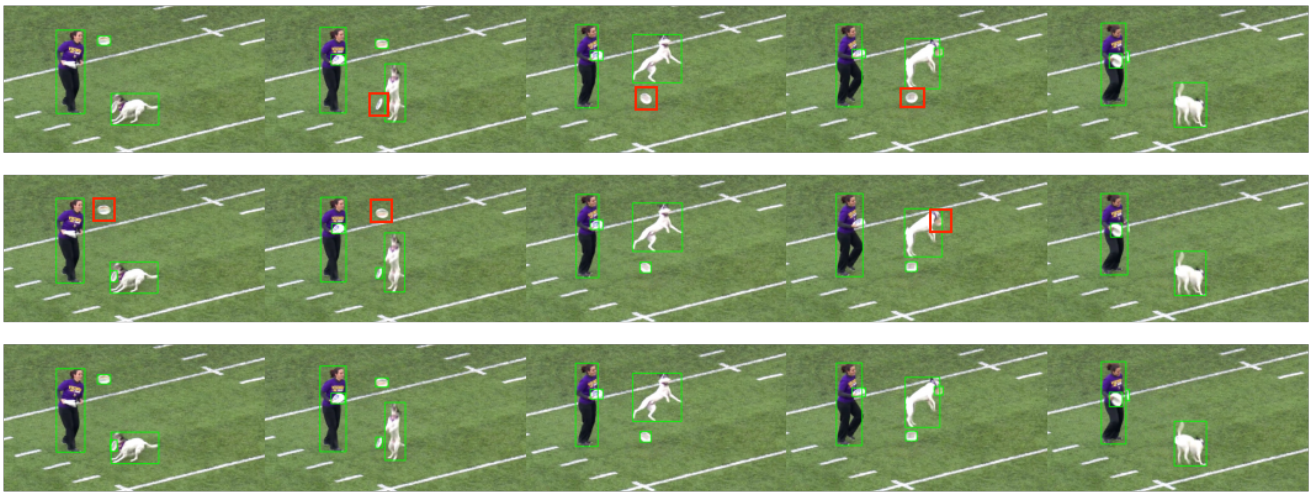


Figure 4. Example of forward tracklets (first row), backward tracklets (second row), and merged tracklets(bottom row). The missing Frisbee discs are marked by red boxes. Forward tracklets and backward tracklets compensate each other by being able to keep track of the objects that the other missed, resulting in the merged tracklets with a better result.

both tracklets. For the same instance, the forward tracklet f and the backward tracklet b may be different. BiTrack module is applied to merge high overlapping tracklets into a final one. Concretely, two tracklets are merged together if the average of IoU between their boxes in valid frames is greater than a threshold thr.

Figure 4 illustrates an example of how a forward tracklets (first row) and a backward tracklets (second row) are merged together into the final result (third row). In the forward tracklets, as seen in the lower half of the images, it is hard to track the Frisbee disc (red box) in the forward path, but easier if doing it backward. Vice versa for another Frisbee disc appearing on the upper half of the images. As a result, two tracklets compensate each other, yielding more robust tracking results.

3.5. Bag of tricks

3.5.1 Multi-task learning

Besides main tasks, we also train the model with the auxiliary tasks, described as follows.

- **Semantic segmentation** The mask head for predicting instance masks only focuses on local information belonging to instances without learning a global concept. Therefore, we suggest adding a semantic segmentation branch to predict masks on a global scale. Specifically, the feature output from the P3 level of FPN is forwarded to stacked convolutions to predict a semantic segmentation mask with 40 channels.

- **Multi-label classification** We propose to add a multi-label classification sub-network to predict categories. Concretely, the backbone feature at the C5 level is fed into the sub-network to predict a 40-class vector. To allow multi-label prediction, we use the Binary Cross Entropy loss during the training.

- **Mask scoring [9]** We further predict the instance segmentation quality in terms of mask IoU. In inference, the mask score is multiplied with the classification score to improve prediction confidence.

3.5.2 Ensemble

We ensemble the predictions of different models into the final results as follows.

- **Detection** We apply Greedy Auto Ensemble [21] to merge predicted boxes of models. Note that, to ensure the merged detection score would be well calibrated, we do not average scores of merged boxes. Instead, we perform the max operation so that the final score would be inherited from the dominant box.

- **Segmentation and Tracking** The bounding boxes obtained from the ensembled models are treated as proposals, which are then fed to different models to extract segmentation mask and embedding representation. Finally, we average masks and embeddings of models to obtain the final ones.

3.5.3 Pseudo label

We take the advantage of the ensemble to generate pseudo labels on the detection of the *valset*. Additionally, we feed these boxes through the tracking module to obtain the most confident ones. Our motivation is that if we can match

Experiments	TCIS	MultiTask	MSD	BiTrack	mAP	$AP50$	$AP75$	$AR1$	$AR10$	$\Delta mAP(\%)$
$A1$					0.309	0.501	0.338	0.269	0.346	-
$A2$	✓				0.331	0.535	0.354	0.285	0.368	2.2
$A3$	✓	✓			0.338	0.546	0.356	0.287	0.374	0.7
$A4$	✓	✓	✓		0.364	0.570	0.402	0.299	0.397	**2.6**
$A5$	✓	✓	✓	✓	**0.388**	**0.589**	**0.438**	**0.320**	**0.436**	2.4

Table 1. Ablation study on the proposed components using the backbone ResNeSt50 on the YoutubeVOS-VIS2021 *valset*. TCIS: Temporal Correlated Instance Segmentation, MultiTask: Multi-task learning, MSD: Multi-Source Data, BiTrack: Bi-directional tracking.

Experiments	Method	mAP	$AP50$	$AP75$	$AR1$	$AR10$
$B1$	S101	0.418	0.652	0.464	0.340	0.454
$B2$	SwinS	0.440	0.666	0.504	0.359	0.476
$B3$	Ensemble (B1, B2)	0.464	0.698	0.515	0.376	0.505
$B4$	S101 + Pseudo	0.539	0.777	0.628	0.421	0.578
$B5$	SwinS + Pseudo	0.560	0.792	0.644	0.430	0.594
$B6$	Ensemble (B1,B2, B4, B5)	**0.575**	**0.806**	**0.671**	**0.441**	**0.609**

Table 2. Ablation study on the bag of tricks with two backbones ResNeSt101 (S101) and SwinS on the YoutubeVOS-VIS2021 *valset*.

boxes over frames, these boxes are likely to represent a foreground object. Thus, they are more reliable than non-trackable ones. Afterward, we combine the *trainset* and the pseudo-label (only detection) *valset* to train new networks.

3.5.4 Label voting

In the tracking-by-detection approach, if the detector misclassifies object labels, the tracking consequently fails to track the object. Consequently, this can break a tracklet into many fragments, and damage the results. To ease this issue, we relax the label consistency criterion. Therefore, detected objects with different categories could be matched together based on their visual embeddings. To this end, a track may still contain different labels, hence, we select the one with highest frequency as the final label for that track.

3.5.5 Multi-scale testing

We utilize Multi-scale testing for further boosting network performance. Alongside the $1\times$ image scale, we also exploit $0.7\times$ and $1.3\times$ scales.

4. Experiments

4.1. Implementation details

All models are trained with Synchronized BatchNorm of batch size 16 on 4 GPUs (4 images per GPU). We use two types of backbone: ResNeSt [22] and SwinTransformer [12]. For ResNeSt backbone, the optimizer is SGD with momentum 0.9 and initial learning rate $1e^{-2}$, while AdamW [13] with initial learning rate $5e^{-5}$ is used for SwinTransformer. Each experiment is trained by 12 epochs, in which, the learning rate is dropped 10 times at the end of epoch 8 and 11. For fast training, we use the image size of 360x640. In our experiments, training with double image size only provides negligible improvement, so this image scale is sufficient. Our code is based on MMDetection [3], and networks are pretrained on the MS COCO 2017 dataset.

4.2. Ablation study

Proposed components At first, we use the model with backbone ResNeSt50 to evaluate the effects of components including Temporally Correlated Instance Segmentation (TCIS), Multi-task learning (MaskScoring, SemSeg, and Multi-label classification), Multi-Source Data (MSD), and Bidirectional Tracking (BiTrack). Table 1 lists results on the YoutubeVOS-VIS2021 *valset*. In Exp. $A1$, we start by a baseline model and achieve 0.309 mAP. By adding TCIS in Exp. $A2$, the metric is improved by 2.2% mAP. Multi-task add-ins (Exp. $A3$) only give a small gain by 0.7% mAP. Then, when applying MSD (Exp. $A4$), the mAP reaches 0.364. Finally, BiTrack (Exp. $A5$) increases the result to 0.388 mAP. These experiments reveal that the three main proposed components constantly leverage the model performance by more than 2% mAP.

Bag of tricks We combine all mentioned techniques in Exp. $A5$ and increase model capacity by training two models with larger backbones ResNeSt101 (Exp. $B1$) and SwinS (Exp. $B2$), yielding 0.418 and 0.440 mAP (see Tab. 2), respectively. By ensembling these two models, we obtain an improved mAP at 0.464 in Exp. $B3$. After generating pseudo data for the detection part in *valset*, we combine the *trainset* and *valset* to re-train the two models and reach boosted performance with mAP 0.539 (Exp. $B4$) and 0.560 (Exp. $B5$). Finally, by ensembling $B1$, $B2$, $B4$, and $B5$, we achieve the state-of-the-art with mAP 0.575.

4.3. Comparison

Method	mAP	$AP50$	$AP75$	$AR1$	$AR10$
tuantng (ours)	**0.575**	**0.806**	**0.671**	**0.441**	**0.609**
eastonssy	0.543	0.792	0.611	0.439	0.588
linhj	0.495	0.727	0.548	0.419	0.591
zfonemore	0.490	0.684	0.548	0.393	0.523
vidit98	0.488	0.694	0.549	0.401	0.550

Table 3. Comparison with other methods on the YoutubeVOS-VIS2021 *valset*.

Method	mAP	$AP50$	$AP75$	$AR1$	$AR10$
tuantng (ours)	**0.541**	0.742	**0.616**	0.433	0.589
eastonssy	0.523	**0.767**	0.577	**0.439**	0.570
vidit98	0.491	0.681	0.545	0.410	0.550
linhj	0.478	0.693	0.527	0.422	**0.591**
hongsong.wang	0.476	0.684	0.529	0.414	0.546

Table 4. Comparison with other methods on the YoutubeVOS-VIS2021 *testset*.

YoutubeVOS-VIS2021 We use the solution in Exp. $B6$ to benchmark on both YoutubeVOS-VIS2021 *valset* and *testset*. Results in Tab. 3 and Tab. 4 show that our method surpasses others by a large margin. Specifically, in the *valset*, our solution achieves 0.575 mAP, which is a large gap of more than 3% to the second method. While transferring to the *testset*, we preserve the first rank to be the State-of-the-art with 0.541 mAP. This indicates that the proposed method has a strong and stable performance on the VIS task. Figure 5 shows sample predictions of our model on the testset. The model can accurately detect categories, segment instance masks, and track objects over frames.

YoutubeVOS-VIS2019 The model used for this benchmark contains backbone SwinS, TCIS, MultiTask, and BiTrack. The result is shown in Tab. 5.

Method	mAP	$AP50$	$AP75$	$AR1$	$AR10$
Ours	**0.543**	**0.766**	**0.656**	**0.470**	**0.579**
MaskProp [2]	0.425	-	0.456	-	-
VisTR [18]	0.401	0.640	0.450	0.383	0.449
CrossVIS [20]	0.366	0.573	0.397	0.360	0.420
CompFeat [7]	0.353	0.560	0.386	0.331	0.403

Table 5. Comparison with other methods on the YoutubeVOS-VIS2019 *valset*. Bold symbols represent the best metrics.

It can be seen that, without MSD, our method can still surpass recent ones with a remarkable gap, i.e., we achieve 0.543 mAP, which is around 11.0% better than the second method (MaskProp). This again demonstrates the effectiveness and generalization of the proposed solution.

5. Conclusion

In this work, we design a unified approach to perform the VIS task within a single model. The success of our solution mainly comes from three proposed components, namely Temporally Correlated Instance Segmentation, Multi-Source Data, and Bidirectional Tracking, as well as applying several practical tricks, e.g. ensemble and pseudo label. Specifically, the three proposed components can boost the performance by approximately 8% mAP with the standard backbone, while the bag of tricks give us a significant improvement by more than 15% mAP with stronger backbones. Leveraging this robust performance, our method outperforms the others significantly and makes new records on the YoutubeVOS VIS 2019 and 2021 datasets.

References

[1] N. Alldrin J. Uijlings I. Krasin J. Pont-Tuset S. Kamali S. Popov M. Malloci A. Kolesnikov T. Duerig A. Kuznetsova, H. Rom and V. Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

[2] Gedas Bertasius and Lorenzo Torresani. Classifying, Segmenting, and Tracking Object Instances in Video with Mask Propagation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 9736–9745, 2020.

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable Convolutional Networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-October:764–773, 2017.

[5] Minghui Dong, Jian Wang, Yuanyuan Huang, Dongdong Yu, Kai Su, Kaihui Zhou, Jie Shao, Shiping Wen, and Changhu Wang. Temporal feature augmented network for video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[6] Qianyu Feng, Zongxin Yang, Peike Li, Yunchao Wei, and Yi Yang. Dual embedding learning for video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[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.

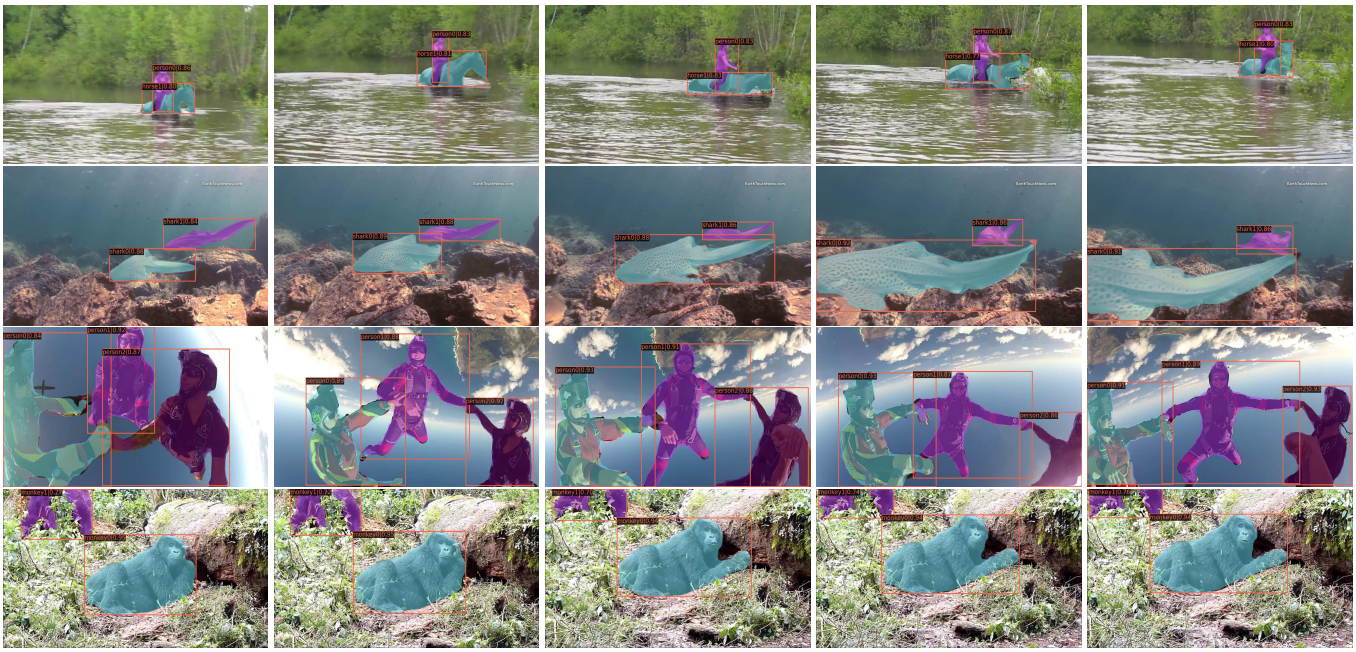


Figure 5. Our model predictions on the YoutubeVOS-VIS2021 *testset*. Each row is a video. The same color represents the same object.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[9] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[10] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[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.

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[14] Jonathon Luiten, Philip Torr, and Bastian Leibe. Video instance segmentation 2019: A winning approach for combined detection, segmentation, classification and tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[15] Sina Mohseni, Mandar Pitale, Jbs Yadawa, and Zhangyang Wang. Self-supervised learning for generalizable out-of-distribution detection. In *AAAI 2020*, 2020.

[16] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2021.

[17] Qiang Wang, Yi He, Xiaoyun Yang, Zhao Yang, and Philip Torr. An empirical study of detection-based video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[18] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2021.

[19] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5188–5197, 2019.

[20] 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.

[21] Yuhang Zang Yan Gao Enze Xie Junjie Yan-Chen Change Loy Yu Liu, Guanglu Song and Xiaogang Wang. 1st place solutions for openimage2019 – object detection and instance segmentation. In *arXiv preprint arXiv:2003.07557*, 2019.

[22] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020.