Video Instance segmentation Challenge 2021 with $YoloV4^{+1}Tr$

Simone Rossetti Temirlan Zharkynbek

Fiora Pirri
Alcor Lab, Sapienza University of Rome
via Ariosto 25, Roma
rossetti.1900592@studenti.uniroma1.it, zharkynbek@diag.uniroma1.it, pirri@diag.uniroma1.it

Abstract

In this report we describe the technical details of the implementation we submitted to the 2021 YouTubeVIS challenge. We propose a novel extension of YoloV4, for video instance segmentation.

1. Introduction

The YouTube VIS dataset proposed in [43] has changed the way of considering video segmentation, by introducing a good number of videos with a significant assortment of situations and scenes and with 40 classes of objects. The diversity of high dynamic video scenes makes the instance segmentation and tracking task quite realistic for several applications and at the same time very challenging.

In recent years, videos interpretation has taken advantage of the introduction of spatio-temporal models with inflated kernels as I3D pretrained on Kinetics [8], which marks a watershed with the shallower 3D ConvNet conjugated with LSTM. I3D and its extensions, such as R (2 + 1) D by [35], and SlowFast by [12] can capture different degrees of temporal variations, as described in [8], though multi-tracking, as in YouTubeVIS, can take little advantage of these models. The reason is that despite dynamic features can help in predicting a subject motion, multi-tracking requires to maximize the matching between the subjects appearing in two frames, in parallel, taking into account the disruption of elements coming in and out of the scene.

Another relevant aspect of multi-tracking in YouTube-VIS is the fact that the transitions are dynamic, as they change scene by scene.

We have submitted to the challenge our proposed instance segmentation model based on YOLOV4 together with a greedy tracker, which we named YOLOV4+1seg, and which we describe in the next sections.

2. Related

Video instance segmentation and tracking is a relatively young research topic clearly assessed in [44]. The task is to identify object instances and their class even if they appear in a single frame, segment and track them throughout the video frames. It requires consistency of the labels when an object instance is occluded and re-appear again after few frames [44]. Although the video instance segmentation task was introduced recently, a lot of work on sub-tasks of it like image instance segmentation, video object tracking, and video semantic segmentation has been done.

Image instance segmentation. The origin of this image and video instance segmentation is the same as both require to group pixels on object instances and semantically classify them [44]. Video instance segmentation, however, also requires to group object instances along all the video frames. Usually, two-stage methods are applied to solve the instance segmentation and classification see, for example, [11, 10, 21, 14]. In our proposed method we also rely on a two stage solution extending YOLOV4 with segmentation.

Video object detection. This task requires detecting objects in video. Recent studies show that spatio-temporal features of consequent video frames improves detection results of individual frames [46, 2, 41, 13]. These metrics focus on single frames without considering instances consistency across frames, which is crucial for tracking.

Video object tracking. There are several tracking related tasks. The semi-supervised task requires a bounding box for each element in the first frame [3, 28, 13]. The unsupervised one requires to detect an object and track it [31, 40, 33]. Even though the latter is quite similar to video instance segmentation, it requires only to detect bounding boxes.

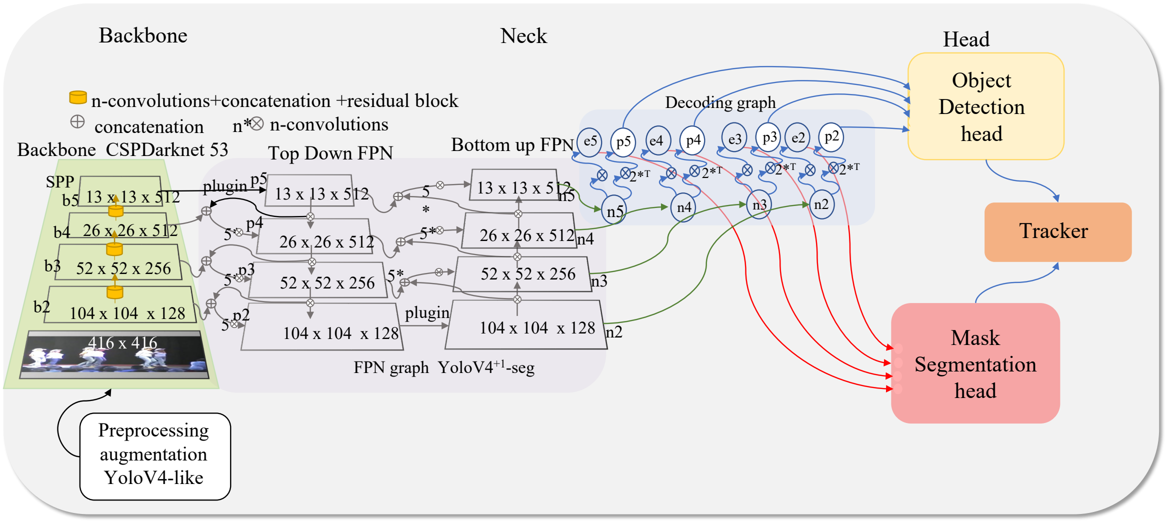


Figure 1. Backbone, FPN and decoding graph of the $YoloV4^{+1}$ model for the instance segmentation

Video semantic segmentation. This task is a direct extension of the image segmentation task requiring the classification of each pixel. Some studies use optical flow as temporal information to improve the performance of semantic segmentation models [47, 22, 32]. Generally, during inference, the ground truth of the first frame is used to propagate motion information to all the video frames. The main difference between this task and the video instance segmentation task is that it does not require to match instances along the video frames.

Video object segmentation. This task requires to segment objects without accounting their class [45, 17, 34] and often it asks to propagate the mask in the first frame to the remaining video frames [45, 17, 34, 36, 30, 7].

This task can be semi-supervised or unsupervised [34, 17]. On the other hand, VIS requires to detect instances, segment and link them along the whole video.

VIS task and our contribution Most of the instance segmentation models use template masks and fine-tune with them [26, 20, 1, 37, 25, 7]. The use of mask make the strong assumption that all the element in the scene appear in the first frame. Similarly, in [9, 16, 29, 42, 45] the masks of the initial frame are used for later comparison, which does not allow to discover new subjects that appear after the first frame. Some approaches to video instance segmentation models use re-detection for tracking. Mainly they use two-stage R-CNN detectors as pairs of Siamese networks [19, 38], but it is not clear on how many classes and instances these methods provide a good accuracy. In our model we do not use prior bounding box or either masks, the additional layer on the neck of the network allows handling VIS task with the high result and applicable real-time speed.

3. Method

As outlined in the introduction we have used for the instance segmentation part an extension of YOLOV4 which we call $YoloV4^{+1}$ and for the multi-tracker we used a greedy association method based on pruning heuristics.

3.1. Instances tracking

This is a simple local method which aims at a fast mitigation of the bad associations which propagates over the tracking sequence. The method is divided in 4 steps:

- Given a sequence of two frames at time t and $t+1$ compute the cost matrix A where each entry from row i (detections at time t) and column j (detections at time $t+1$) is the class+mask IOU_j^i score.
- Gating: check for associations which score is lower than a fixed threshold.
- Best Friend: select only associations which are minimum of both rows and columns.
- Lonely Best Friend: select only associations whose difference of minimum and second minimum is greater than a fixed threshold.

3.2. Instances segmentation

The proposed model is based on the YoloV4 architecture [4] for detection, which we extend to cope with instance segmentation. The choice of YoloV4 [4] is natural for tracking, as YoloV4 is designed for fast detection.

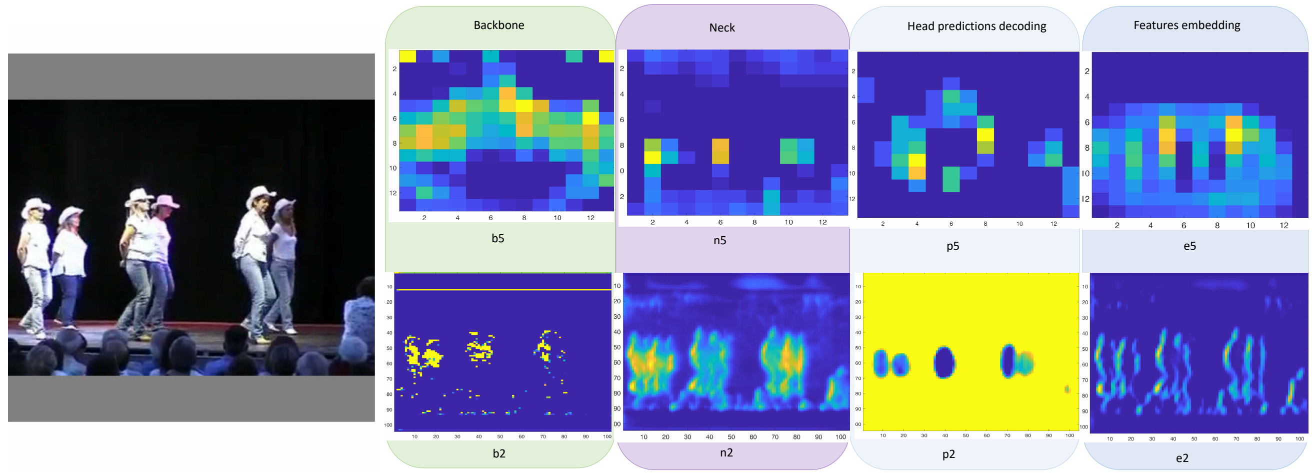


Figure 2. Feature maps of the upper (first row) and lower (second row) level of the pyramid, from the shallower (first column) to the deeper output layer (last column). Notice that bigger objects are encoded at deeper level, while the smallest ones do no survive at the down-sampling.

YoloV4 baseline architecture is formed by a backbone, a neck and a detection head. According to the authors [4] CSPDarknet53 (CSP is the acronym of Cross Stage Partial Network) is the best backbone on COCO dataset [23]. CSPDarknet53[4] is topped by a Spatial Pyramid Pooling (SPP) allowing to plug-into the feature pyramid network (FPN). The SPP had been adapted from [15] already in YoloV3 by concatenating max-pooling outputs, with the effect of increasing the receptive fields.

In the proposed model we adopt CSPDarknet53, with Mish activation [27] expanding it with a layer. In fact, we expand the FPN with a layer too, both in top-down and bottom-up pathways, with lateral connections, namely the elements b_2, p_2 and n_2, see Figure 1. Due to the limit of YoloV4 on small objects. Features from larger maps extending the range from coarse to fine in the bottom-up and top-down pathways improve localization [5, 6] together with the aggregating step carried out by the lateral connections, see Figure 1. Despite this extension of the pyramid is crucial for instance segmentation, its drawback is the slowing down of the detection head, as we carry on all the layers also to the proposal graph. Here we introduce concatenation to aggregate layers, as opposed to PANet [24], which adopts addition. Each block of our FPN, differently from PANet [24], is structured as in YoloV4, though extended. Namely, our FPN aggregates the reconstructed layer with the corresponding feature map after one convolution, an up-sampling (resp. down-sampling) and a second convolution with the same one dimensional kernel. After the concatenation, a kernel of size 1 is alternated with a kernel of size 3 doubling the channels dimension. This induces dilation and contraction of features. In PANet [24], on the other hand, the 3 dimensional kernels are added all to the bottom of the top-down pathway, while in the bottom-up the kernels have all size 3 and the same number of channels, tailoring the FPN to large images. The results of our strategy on the feature maps can be seen in Figure 2.

YoloV4 [4] detection head uses fixed ratios for the anchors, which are given as priors, actually computed in advance by Kmeans, and uses the predictions from the FPN encoded as rough proposals, including their confidence. With similar perspective for the mask prediction, we have created an encode-decode-graph, which separates the predictions from the features embeddings to be used in the ROI pooling for each target segmentation.

Generating embeddings for features segmentation was also done in [39], though here [39] the authors use the embeddings to capture the features of each subject identity in the whole dataset, which is an impossible endeavor for a huge dataset with hundreds of thousands identities.

The predictions, for the detection head, are encoded by convolving each pyramid layer with kernels of size one and three, taking into account the anchors ratios, and suitably reshaping and transposing the obtained tensors. Figure 2 show the features out-coming from the backbone, from the FPN and the encode-decode graph.

For the detection head we actually follow YoloV4 main structure in our proposals graph and ROIAlign. The predictions are tensors of size $nB \times nA \times nw \times nh \times nC$, where nB is the batch size (8 for us) nA is the number of anchors for each pyramid layers (which strongly depends on the dataset, e.g. for YouTubeVIS21 10 clusters for each of the 4 layers give a reasonable optimal mAP of 0.8), nW and nH specify the pyramid layer size and nC are the principal channels: 4+1 for the regression and confidence box, k for the classification according to the number of classes. Tensors are decoded first by separating the confidence, which is transformed into a probability by

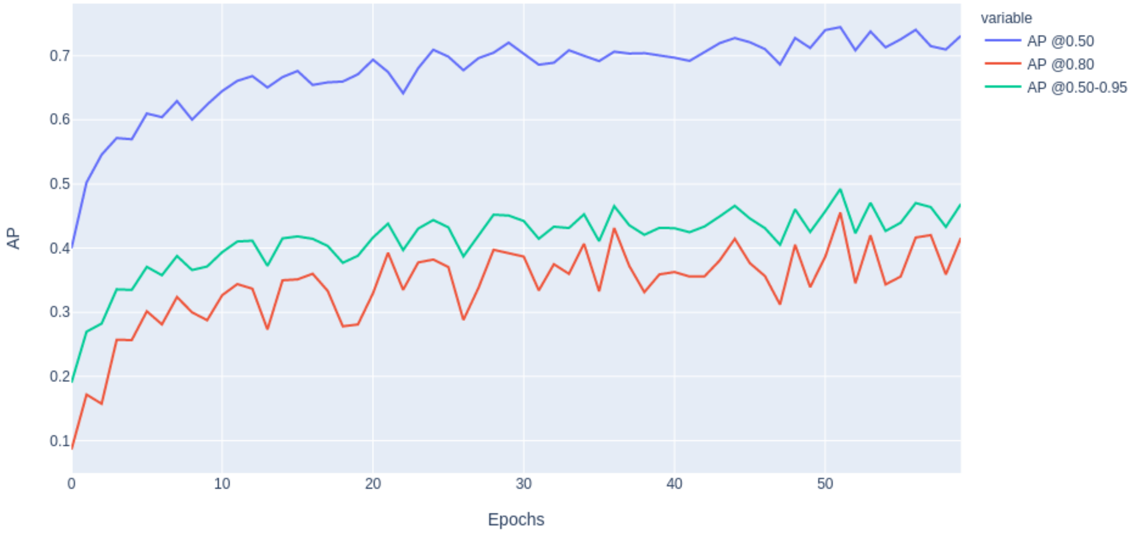


Figure 3. Instance segmentation Average Precision Results on YouTubeVIS21 at 0.5, 0.8 and 0.5-0.95 mAP, same metrics used for COCO instance segmentation dataset, which takes into account segmentation IOU and classification.

softmax, and then by computing the deviation from the ground truth as a δ-map with respect to a fixed grid. The δ-map is used to learn the displacement of the proposals from ground truth.

The decoded proposals are selected by consistency (size and proportion) and score, using regular non-max suppression algorithm; first for each level of the pyramid, then on all levels, in order to guarantee higher variety and less impact on device memory. Computing IOU between proposals and ground truth we can assign the relative instance to be segmented. At the same time ROI Pooling takes place, in particular we have adopted the same strategy used in PANet, namely Adaptive Feature Pooling, in which using the proposals we cut and resize the regions of interest on the embeddings produced by the four levels of the pyramid. These new features are fed to a different convolutional subnet which learn instance segmentation. MaskRCNN and similiar approaches makes use of a third subnet which is trained to predict classes and deviation of the proposals from the ground truth.

Joint Losses The multi-task nature of the network requires the adoption of four different losses, one for each head. As stated before bounding box regression is performed minimizing the Smooth L1 Loss with respect to the encoded anchors deviation from the ground truth. Proposal confidence and segmentation optimization follows the standard binary cross entropy loss with sigmoid, while a cross entropy loss with softmax is used to optimize the multi-class classification problem. Each class is encoded at the centroid of each instance, while the surrounding area is unlabeled in order to prevent the injection of noise in case of nearby classes.

The joint of the learning objectives of each head is performed by the adoption of a weighting mechanism named *Automatic Loss Balancing*, namely the version proposed here [18]. Which is nothing more than a weighted sum of the task losses:

$$\mathcal{L}_{tot} = \sum_{i}^{M} \sum_{j}^{N} \frac{1}{2} \left(\frac{1}{e^{w_j^i}} \mathcal{L}_j^i + w_j^i \right) \tag{1}$$

with M the number of pyramid layers, N the number of the considered loss and w_j^i the learnable weighting coefficient of the $i-th$ layer and the $j-th$ loss function.

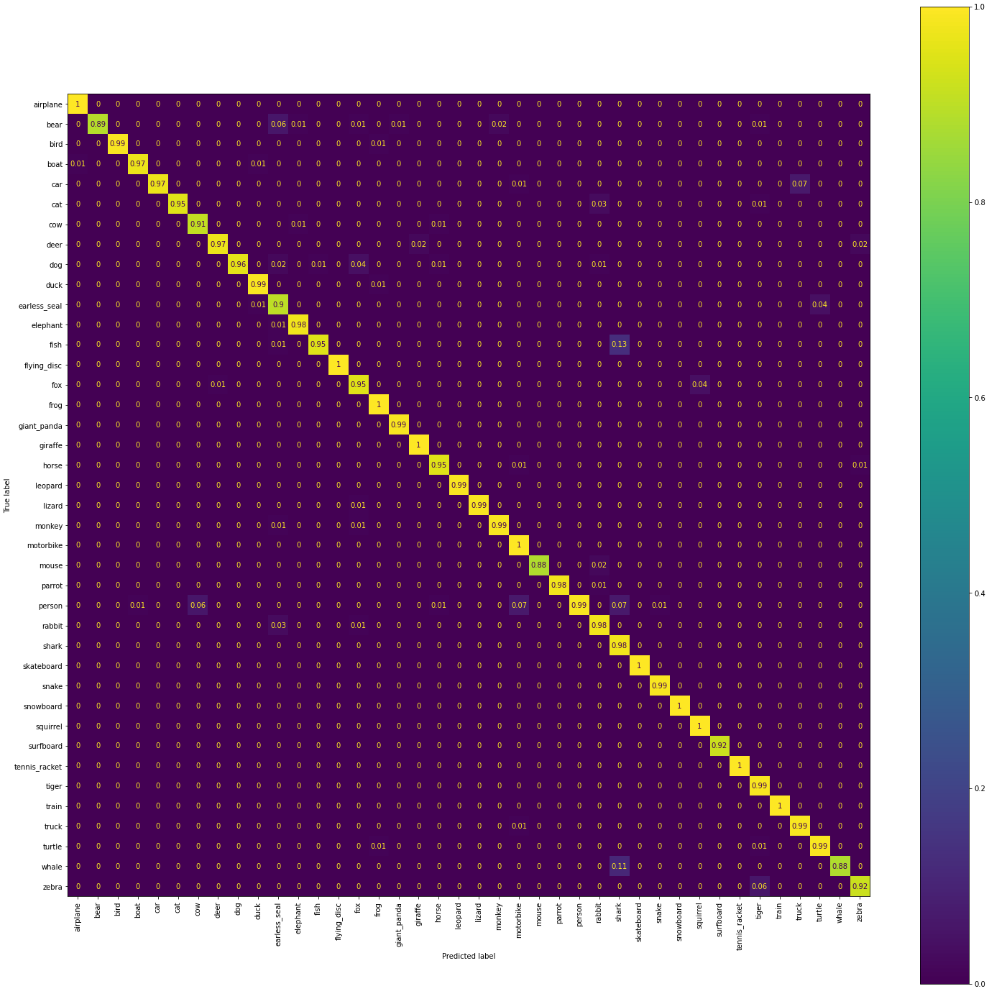
4. Implementation details

The method is implemented in Tensorflow 2.3

5. Results

5.1. Results of the instance segmentation

Results are illustrated in Figure 3 showing the average precision and in Figure 4 showing the confusion matrix for the 40 classes.



	airplane	bear	bird	boat	car	cat	cow	deer	dog	duck	earless_seal	elephant	fish	flying_disc	fox	frog	giant_panda	giraffe	horse	leopard
airplane	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
bear	0	0.89	0	0	0	0	0	0	0	0	0.06	0.01	0	0	0.01	0	0.01	0	0	0
bird	0	0	0.99	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0
boat	0.01	0	0	0.97	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0
car	0	0	0	0	0.97	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cat	0	0	0	0	0	0.95	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cow	0	0	0	0	0	0	0.91	0	0	0	0	0.01	0	0	0	0	0	0	0.01	0
deer	0	0	0	0	0	0	0	0.97	0	0	0	0	0	0	0	0	0	0.02	0	0
dog	0	0	0	0	0	0	0	0	0.96	0	0.02	0	0.01	0	0.04	0	0	0	0.01	0
duck	0	0	0	0	0	0	0	0	0	0.99	0	0	0	0	0	0.01	0	0	0	0
earless_seal	0	0	0	0	0	0	0	0	0	0.01	0.9	0	0	0	0	0	0	0	0	0
elephant	0	0	0	0	0	0	0	0	0	0	0.01	0.98	0	0	0	0	0	0	0	0
fish	0	0	0	0	0	0	0	0	0	0	0.01	0	0.95	0	0	0	0	0	0	0
flying_disc	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
fox	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0.95	0	0	0	0	0
frog	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
giant_panda	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.99	0	0	0
giraffe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
horse	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.95	0
leopard	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.99
lizard	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0
monkey	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0.01	0	0	0	0	0
motorbike	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mouse	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
parrot	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
person	0	0	0	0.01	0	0	0.06	0	0	0	0	0	0	0	0	0	0	0	0.01	0
rabbit	0	0	0	0	0	0	0	0	0	0	0.03	0	0	0	0.01	0	0	0	0	0
shark	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
skateboard	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
snake	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
snowboard	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
squirrel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
surfboard	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
tennis_racket	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
tiger	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
train	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
truck	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
turtle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0
whale	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
zebra	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	lizard	monkey	motorbike	mouse	parrot	person	rabbit	shark	skateboard	snake	snowboard	squirrel	surfboard	tennis_racket	tiger	train	truck	turtle	whale	zebra
airplane	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
bear	0	0.02	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0
bird	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
boat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
car	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0.07	0	0	0
cat	0	0	0	0	0	0	0.03	0	0	0	0	0	0	0	0.01	0	0	0	0	0
cow	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
deer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02
dog	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0
duck	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
earless_seal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.04	0	0
elephant	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
fish	0	0	0	0	0	0	0	0.13	0	0	0	0	0	0	0	0	0	0	0	0
flying_disc	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
fox	0	0	0	0	0	0	0	0	0	0	0	0.04	0	0	0	0	0	0	0	0
frog	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
giant_panda	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
giraffe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
horse	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01
leopard	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
lizard	0.99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
monkey	0	0.99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
motorbike	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mouse	0	0	0	0.88	0	0	0.02	0	0	0	0	0	0	0	0	0	0	0	0	0
parrot	0	0	0	0	0.98	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0
person	0	0	0.07	0	0	0.99	0	0.07	0	0.01	0	0	0	0	0	0	0	0	0	0
rabbit	0	0	0	0	0	0	0.98	0	0	0	0	0	0	0	0	0	0	0	0	0
shark	0	0	0	0	0	0	0	0.98	0	0	0	0	0	0	0	0	0	0	0	0
skateboard	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
snake	0	0	0	0	0	0	0	0	0	0.99	0	0	0	0	0	0	0	0	0	0
snowboard	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
squirrel	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
surfboard	0	0	0	0	0	0	0	0	0	0	0	0	0.92	0	0	0	0	0	0	0
tennis_racket	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
tiger	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.99	0	0	0	0	0
train	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
truck	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0.99	0	0	0
turtle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0.99	0	0
whale	0	0	0	0	0	0	0	0.11	0	0	0	0	0	0	0	0	0	0	0.88	0
zebra	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.06	0	0	0	0	0.92

Figure 4. Confusion Matrix over 40 classes of YouTubeVIS21 dataset.

6. Acknowledgments

We really thank the YouTubeVIS team for the very well organized challenge, for the workshop and for the dataset.

References

[1] L. Bao, B. Wu, and W. Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5977–5986, 2018. 2

[2] G. Bertasius, L. Torresani, and J. Shi. Object detection in video with spatiotemporal sampling networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 331–346, 2018. 1

[3] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016. 1

[4] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 2, 3

[5] P. J. Burt. Smart sensing within a pyramid vision machine. *Proceedings of the IEEE*, 76(8):1006–1015, 1988. 3

[6] P. J. Burt and E. H. Adelson. Merging images through pattern decomposition. In *Applications of Digital Image Processing VIII*, volume 575, pages 173–181. International Society for Optics and Photonics, 1985. 3

[7] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017. 2

[8] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4724–4733. IEEE, 2017. 1

[9] J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang. Fast and accurate online video object segmentation via tracking parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7415–7424, 2018. 2

[10] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3150–3158, 2016. 1

[11] R. Faster. Towards real-time object detection with region proposal networks shaoqing ren [j]. *Kaiming He, Ross Girshick, and Jian Sun*, 2015. 1

[12] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6202–6211, 2019. 1

[13] C. Feichtenhofer, A. Pinz, and A. Zisserman. Detect to track and track to detect. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3038–3046, 2017. 1

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[15] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015. 3

[16] Y.-T. Hu, J.-B. Huang, and A. G. Schwing. Videomatch: Matching based video object segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 54–70, 2018. 2

[17] S. D. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 2117–2126. IEEE, 2017. 2

[18] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, 2018. 4

[19] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. ˇCehovin Zajc, T. Vojir, G. Bhat, A. Lukezic, A. Eldesokey, et al. The sixth visual object tracking vot2018 challenge results. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2

[20] X. Li and C. C. Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 90–105, 2018. 2

[21] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2359–2367, 2017. 1

[22] Y. Li, J. Shi, and D. Lin. Low-latency video semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5997–6005, 2018. 2

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 3

[24] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018. 3

[25] J. Luiten, P. Voigtlaender, and B. Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *Asian Conference on Computer Vision*, pages 565–580. Springer, 2018. 2

[26] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. Video object segmentation without temporal information. *IEEE transactions on pattern analysis and machine intelligence*, 41(6):1515–1530, 2018. 2

[27] D. Misra. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681*, 2019. 3

[28] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4293–4302, 2016. 1

[29] S. W. Oh, J.-Y. Lee, K. Sunkavalli, and S. J. Kim. Fast video object segmentation by reference-guided mask propagation.

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7376–7385, 2018. 2

[30] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 2

[31] A. Sadeghian, A. Alahi, and S. Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 300–311, 2017. 1

[32] E. Shelhamer, K. Rakelly, J. Hoffman, and T. Darrell. Clockwork convnets for video semantic segmentation. In *European Conference on Computer Vision*, pages 852–868. Springer, 2016. 2

[33] J. Son, M. Baek, M. Cho, and B. Han. Multi-object tracking with quadruplet convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5620–5629, 2017. 1

[34] P. Tokmakov, K. Alahari, and C. Schmid. Learning motion patterns in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3386–3394, 2017. 2

[35] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 1

[36] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe. Mots: Multi-object tracking and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7942–7951, 2019. 2

[37] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. *arXiv preprint arXiv:1706.09364*, 2017. 2

[38] P. Voigtlaender, J. Luiten, P. H. Torr, and B. Leibe. Siam r-cnn: Visual tracking by re-detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6578–6588, 2020. 2

[39] Z. Wang, L. Zheng, Y. Liu, and S. Wang. Towards real-time multi-object tracking. *arXiv preprint arXiv:1909.12605*, 2(3):4, 2019. 3

[40] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017. 1

[41] F. Xiao and Y. J. Lee. Video object detection with an aligned spatial-temporal memory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 485–501, 2018. 1

[42] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 585–601, 2018. 2

[43] L. Yang, Y. Fan, and N. Xu. Video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1

[44] L. Yang, Y. Fan, and N. Xu. Video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5188–5197, 2019. 1

[45] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. Efficient video object segmentation via network modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6499–6507, 2018. 2

[46] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 408–417, 2017. 1

[47] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei. Deep feature flow for video recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2349–2358, 2017. 2