# Zero-Shot Video Object Segmentation with Salience Masks

Gregory Shikhman
shikhman@stanford.edu

## Abstract

*This paper presents an unsupervised Video Object Segmentation algorithm that incorporates an object salience heuristic. The algorithm is a modification of the DAVIS 2020 competition winner FrameSelect. The FrameSelect algorithm is analyzed and we qualitatively identify weaknesses in the salient object selection process based on examination of the output predictions within a validation dataset. We evaluate several approaches for improving object selection and select an approach incorporating a detector based on classical computer vision techniques called Robust Background Detection. The detector output is used to weight candidate masks and exclude low-salience masks. An experiment is conducted varying the mask salience threshold used and results are compared using the Jaccard Index and Contour Accuracy figures of merit defined by the DAVIS 2020 competition. The mask salience heuristic does not improve performance, but some analysis is done to examine why. Further improvements are proposed to augment mask selection in the unsupervised Video Object Segmentation problem by considering it as a probabilistic tracking problem.*

## 1. Introduction

Video object segmentation (VOS) is a process to label regions in a video into distinct objects, maintaining the labeling over the course of the video [8]. VOS is a key building block for robotics, augmented reality, and other disciplines where scene understanding is a prerequisite for further processing.

The Unsupervised VOS (UVOS) formulation is a particularly new and challenging VOS problem [1]. The most commonly studied variant of VOS so far has been sipervised/one-shot VOS, which relies on an initial human labeling at the start of a video to bootstrap the tracking. One-shot simplifies the problem as the supervised labeling usually excludes background objects and only tracks the most salient objects within the scene. UVOS is harder because the process must also decide on salient objects to track over the course of the video.

The key challenges to solve in UVOS are salience, tracking and permanence [3]. The salience problem as described above requires identifying the most relevant objects as a human would recognize them. Tracking means following specific object instances (e.g. a specific basketball player in a basketball game) despite changes in pose, similar looking object instances, and occlusions. Permanence is specifically about tracking an object despite partial or complete occlusion by other objects in the video or the edge of the video frame.

In this work we build upon a prior UVOS method called FrameSelect [3][10], which bootstraps masks to track using Mask R-CNN [4], spatially and temporally tracks them throughout a video using self attention [7], and updates for mask drift with conditional re-initializations of the Mask R-CNN object mask based on the results of a classification network called Selector Net.

We analyze deficiencies in the segmentation from a dataset of challenging scenarios containing occlusions, rapid motion, and many object instances. We then identify a weakness in object salience, likely due to poor mask selection from the Mask R-CNN candidates due to sparse training data.

We incorporate a new mask selection process based on Robust Background Detector (RBD) [13], an algorithm based on classical computer vision approaches, and evaluate our new pipeline using the DAVIS evaluation criteria and our own qualitative analysis.

## 2. Related Work

We build upon the FrameSelect method which was the top scoring method (as measured by J and F scores) in the DAVIS 2020 Unsupervised challenge. This method uses a combination of mask evolution using unsupervised self-attention with a Space-Time Memory (STM) network, mask generation with Mask R-CNN, and mask evaluation with hand-crafted features and a supervised fully-connected network.

The components of FrameSelect work to compliment each other. STM is the key VOS algorithm but was built for supervised VOS and does not create mask proposals required for UVOS. Mask R-CNN creates high quality mask proposals but does not use temporal data, so it does extremely poorly with the tracking problem of VOS. The hand-crafted features and neural network mask evaluator connect the two mask sources and optimize them for each frame.

The hand-crafted feature of FrameSelect measures region similarity (similar to the J metric although not normalized for mask size) between a candidate mask and the mask from the last frame, and is combined with a supervised fully-connected network to decide whether to integrate an evolved STM mask or a new Mask R-CNN proposal mask into the current frame. The region similarity feature seems biased towards including large mask candidates as it is not normalized like the J metric, and the Mask R-CNN proposal has no temporal data and is vulnerable to tracking and occlusion errors between frames. The Mask R-CNN proposals are also just the top 10 most confident instance/mask proposals in the frame over a 10% confidence threshold, which is extremely sensitive to non-salient but highly confident detections like bystanders in sporting events and household objects in indoor settings.

The STM method for mask evolution is from the 2019 DAVIS competition, in the ”semi-supervised” track which provides initial image labelings. The method two deep neural encoders, a ”memory encoder” and a ”query encoder”, to encode spatio-temporal mask data and retrieve it from a key-value store respectively. The network is run in a ”forward” direction where the initial labeled frame is fed into the encoder, then the query encoder encodes the second (unlabeled) frame, and the decoder predicts the evolved object masks from the encoded initial frame. This process is repeated for each frame in the video, with the look-behind of the decoder relying on a buffer of encoded past images combined with a ”Space-Time Memory Read” which selects the most relevant past encoded image from the buffer. The Space-Time Memory Read algorithm and the key generated by the memory and query encoders are the self-attention mechanism of the method.

The STM algorithm has an interesting hyper-parameter which was not well-explored in 2019 competition. Ideally, all past frames would be stored and evaluated to find the right frame to decode from, but due to memory constraints and the authors' goal of creating a fast real-time VOS method, the reference implementation only stores and evaluates every 5th frame of the video. The authors report an absolute improvement of 13% using every 5th frame compared to just using the initial frame of the video, and an absolute improvement of 10% using the last frame plus the first frame of the video. DAVIS video sequences are almost all 60 frames are less, so it is possible that discarding 80% of frames hurts STM performance, particularly for fast-changing videos.

Another interesting alternative or compliment to using R-CNN masks to generate candidate masks is to incorporate optical flow data. This idea is mature, with numerous approaches dating back over two decades. A particularly interesting and simple approach is computing relative motion of objects from cumulative optical flow [11] through a video sequence and selecting objects with high relative motion. This method should be robust to camera motion and does not make strong assumptions about the qualities of salient objects besides that they should have high relative motion in a video sequence. Frame-to-frame motion from optical flow has has also been directly used in the Video Object Segmentaton task to compute salient masks, however, under an ablation study [2], the optical flow data only improved performance by 3% relative to just relying on R-CNN segmentation.

## 3. Approach

### 3.1. Qualitative Analysis Tool

We have created a visualization tool to qualitatively analyze VOS predictions from the baseline FrameSelect algorithm. The tracking and mask evolution of the algorithm seems excellent, however, it often tracks extraneous objects in the foreground/background (see Figures 2 and 3). From our visualizations we decided to add additional means of saliency detection to the FrameSelect pipeline to remove extraneously tracked objects. There are also issues with mask tracking during dramatic changes in pose (see Figure 4) which may be addressed by tuning the STM lookback hyper-parameter, but we did not explore this yet.

### 3.2. Augmenting FrameSelect with Salience

In our proposed improvement to FrameSelect we incorporate the Robust Background Detection (RBD) method [13]. This method maximizes a heuristic called the Boundary Connectivity which is an ratio of how many pixels in an image region (arbitrarily chosen) on the image boundary/edge to how many pixels are in the image region. Computing Boundary Connectivity of arbitrary combinations of pixels is not computationally efficient, so the image is first segmented into superpixels, which are k-means clusters of pixels. These superpixel boundary connectivity values are combined with contrast and smoothness heuristics and optimized with a least-squares method to create a saliency mask with foreground objects highlighted.

The grouping, contrast, and smoothness heuristics incorporated in the RBD algorithm seem to work well for single foreground images (see Figure 5). However, the DAVIS dataset is intentionally constructed with challenging scenes, and the results shown do not necessarily generalize over other more challenging samples from the data.

We use RBD to improve the initial mask selection and new candidate mask selection. A salience heuristic is com-

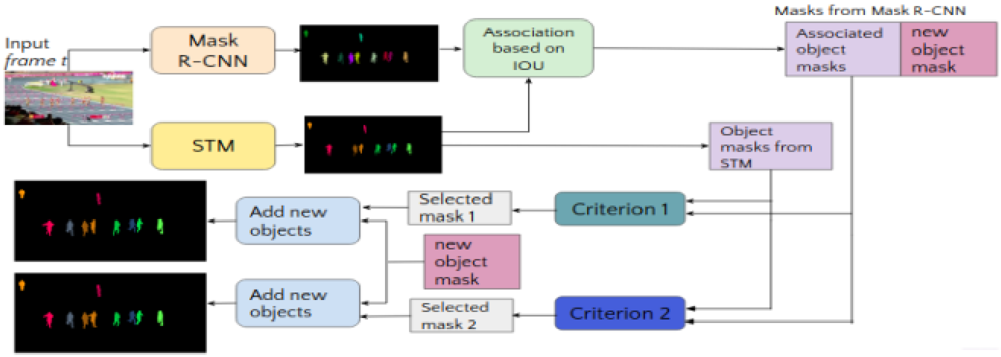

Figure 1. FrameSelect block diagram (Garg et. al 2020)

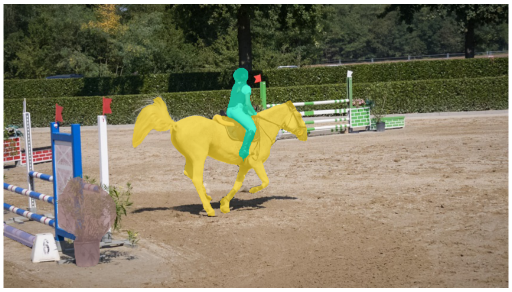

Figure 2. High-confidence but low-salience plant masked in foreground.

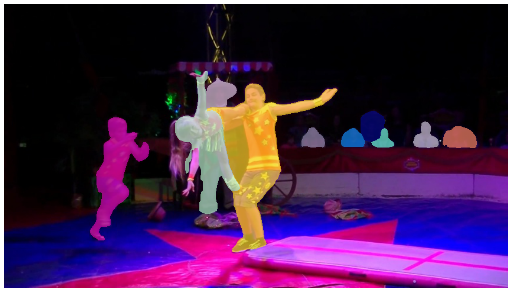

Figure 3. High-confidence but low-salience audience members in background.

puted based on the average RBD salience value over a detected R-CNN mask, and the heuristic value is compared to a fixed threshold to winnow the candidate masks provided by Mask R-CNN. This should improve the J metric and qualitatively reduce the number and size of false positive segmentations. The threshold is a tuning parameter and we examine two different thresholds (0.5 and 0.95) to see if there is any difference in performance.

The salience heuristic is calculated as follows, where $M$

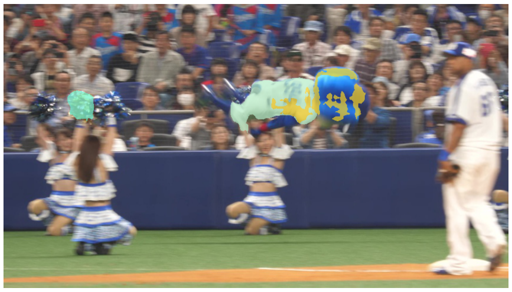

Figure 4. Fast-changing pose of the person in the middle causes a poorly evolved mask.

is a candidate mask, and $p$ is a pixel in the mask:

$$S(M) = \frac{\sum_{p \in M} \mathrm{rbd}(p)}{|M|}$$

and the mask is included if $S(M) > S_{\text{threshold}}$. RBD masks are calculated for all DAVIS data in a preprocessing step.

As part of our work, we have partially reproduced the FrameSelect results. Although the FrameSelect implementation is available on GitHub, it is not actually provided in a runnable state. First, the codebase first expects that DAVIS data is pre-processed with a Mask R-CNN implementation which is not provided. We have created a small pipeline to create Mask R-CNN segmentation images from the input DAVIS data using the Detectron2 library [12]. The specific Mask R-CNN model used was the Feature Pyramidal Network (FPN) backbone trained on the COCO model (Detectron2 model id 137849458), with a minimum detection confidence threshold of 10% maximum of 10 objects detected per frame.

For our RBD augmentation of FrameSelect we have used an RBD implementation from GitHub with some modifications [5]. We create a script to transform the DAVIS data into RBD saliency fields as seen in Figure 5. The saliency

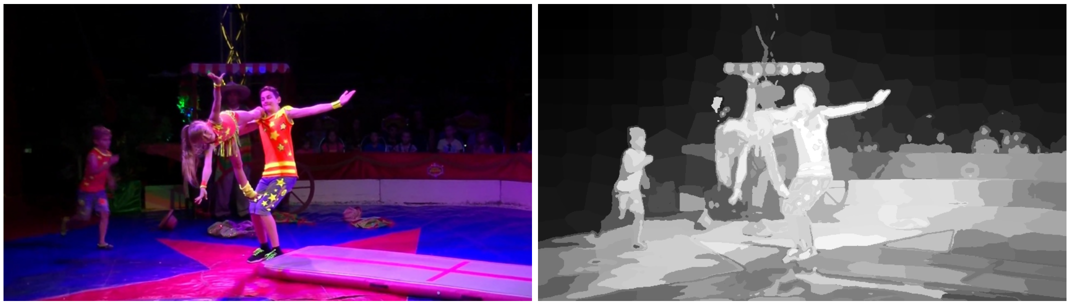

Figure 5. Sample output from RBD preprocessing. High-salience pixels are brighter.

field was normalized so that minimum saliency had 0 intensity and maximum saliency had 255 intensity.

The RBD data was fed into the modified FrameSelect algorithm. Two values $S_{\text{threshold}}$ were tested for thresholding salient masks, 0.5 and 0.95.

| Method | J | F |
|---|---|---|
| FrameSelect (Orig) | 52.9 | 63.3 |
| FrameSelect (Ours) | 49.2 | 59.5 |
| FrameSelect+$S_{0.5}$ | 48.6 | 59.0 |
| FrameSelect+$S_{0.95}$ | 48.0 | 58.0 |
| UnVOST | 54.0 | 62.0 |
| KIS | 50.0 | 58.3 |

Table 1. Results on the DAVIS 2020 Unsupervised test-dev dataset.

# 4. Results

## 4.1. Dataset Information

We participate in the Unsupervised Video Object Segmentation challenge of the Densely Annotated VIdeo Segmentation (DAVIS) 2020 competition [1]. The competition provides 60 training, 30 validation, and 2x30 test sequences of RGB 854x480 image stills from videos. No camera calibration data is provided. The training and validation sets provide ground-truth human labeled annotations, whereas the two test sequences have hidden ground-truth data which is only used by the competition for evaluation on the CodaLab platform. Although the competition leaderboard is now closed to new entrants, we can still submit our results for grading on CodaLab.

## 4.2. Measurements Methodology

We use the DAVIS evaluation metrics [1] for assessing our algorithm's performance: the region-based segmentation similarity Jaccard index (J), and the contour accuracy (F). The Jaccard index measures overlap by taking the number of pixels in the intersection between the predicted $M$ and true mask $G$, and dividing it by the union of the two (so the maximum value is for a complete intersection of identical masks): $J = \frac{M \cap G}{M \cup G}$. The contour accuracy measures the precision and recall between contour (exterior) points of the predicted and true masks with bipartite graph matching ($P_C$ and $R_C$ respectively), and computes the ratio $F = \frac{2 P_C R_C}{P_C + R_C}$.

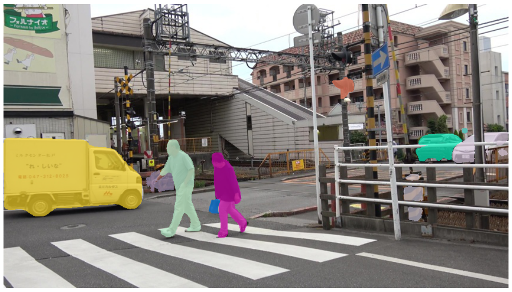

Figure 6. Overlay on input image.

These metrics are computed across all objects in each frame of each sequence and averaged across the sequence. More formally, for a metric $M$ and a video sequence $S$, the mean performance $m(M, S)$ is:

$$m(M, S) = \frac{1}{|O_s|} \sum_{o \in O_S} \frac{1}{|F_{s(o)}|} \sum_{f \in F_{s(o)}} M(m_o^f, g_o^f)$$

where $O_S$ is the total set of objects in the sequence, $o$ is an object in $O_S$, $F_{s(o)}$ is the frames containing $o$, and $m_o^f, g_o^f$ are the predicted and true masks respectively of that object in that frame. The mean $J$ and $F$ are the main metrics for each sequence. The DAVIS authors provide an implementation of the evaluation algorithms, which we have graciously used [9].

We also qualitatively evaluate the segmentation performance by examining false-positive and false-negative masks from overlays of the predicted masks against the input sequences using a visualization script that we have created (Figure 6).

We have quantitatively measured the performance of the baseline FrameSelect implementation and the implementaton with our saliency heuristic using the CodaLab evaluation server. As mentioned previously, the CodaLab server was set up to facilitate the DAVIS competitions but is still open to evaluate new implementatons. Via the CodaLab server, we evaluate the J and F figures of merit on a test dataset for which the ground truth data is hidden from the public.

We also specifically examine the performance in several training sequences that were qualitatively identified as performing poorly with the baseline FrameSelect implementation. The performance is evaluated using given ground-truth data and using the J and F evaluation code provided by the DAVIS organizers [9].

| Method | Sequence | J | F |
|---|---|---|---|
| FrameSelect (Ours) | HorseJumpLow | 85.2 | 96.0 |
| FrameSelect+$S_{0.5}$ | HorseJumpLow | 85.2 | 97.0 |
| FrameSelect+$S_{0.95}$ | HorseJumpLow | 85.2 | 97.0 |
| FrameSelect (Ours) | Sheep | 78.0 | 86.9 |
| FrameSelect+$S_{0.5}$ | Sheep | 78.5 | 87.1 |
| FrameSelect+$S_{0.95}$ | Sheep | 78.5 | 87.1 |
| FrameSelect (Ours) | DogGooses | 76.8 | 89.5 |
| FrameSelect+$S_{0.5}$ | DogGooses | 76.8 | 89.5 |
| FrameSelect+$S_{0.95}$ | DogGooses | 76.8 | 76.8 |
| FrameSelect (Ours) | Butterfly | 0.1 | 0.02 |
| FrameSelect+$S_{0.5}$ | Butterfly | 63.2 | 71.5 |
| FrameSelect+$S_{0.95}$ | Butterfly | 63.2 | 71.5 |

Table 2. Results on the DAVIS 2020 Unsupervised training and test sets for select sequences.

## 5. Conclusion

Our salience heuristic for incorporating new masks did not improve mean performance compared to the baseline FrameSelect implementation on the DAVIS test set. The heuristic we implemented relies on contrast and a ratio of area to edge perimeter and does not work well on the DAVIS dataset which contains sequences specifically designed to frustrated simple approaches with confounding foreground objects. In specific sequences that fit our assumptions of a simple foreground/background like the Butterfly and HorseJumpLow sequences, the RBD heuristic is indeed able to achieve improvements, sometimes ones which are dramatic.

This corresponds with the results of [2] which used optical flow data to augment segmentation masks and also achieved a very minimal average improvement in performance relative to their segmentation baseline as discovered in ablation. Foreground objects still sometimes confuse our RBD heuristic. Setting the threshold high enough is able to decrease false positives in some sequences (see Figure 7 and Table 2) but also sometimes misses true positives across the entire DAVIS dataset. In aggregate the effect is slightly detrimental.

It is also worth considering if the measurement methodology is to blame and not the algorithm (to hate the sin and not the sinner, so to speak). It seems that the $J$ metric is much more sensitive to false negatives than to false positives, as seen in the performance difference on HorseJumpLow vs. Butterfly, where there is a slight performance improvement from excluding a false positive and a huge performance improvement from including a falsie negative. This follows naturally from the defintion of the $J$ metric, where true positive data only appears in the numerator, and hence false negatives always have a zero numerator. In a follow up experiment, it would be worth exploring other figures of merit which are more sensitive to false positives.

Setting aside issues of measurement methodology, there are several deficiencies that could be addressed to improve the performance of our algorithm and of FrameSelect as a whole. First, although our algorithm affects when new masks are included in the sequence to be tracked, masks are never dropped from tracking. It is likely that accuracy cumulatively decreases over sequence length as more non-salient objects are tracked over the course of a sequence. This could be explored in a subsequent experiment and used as motivation for more sophisticated mask management.

Second, and more importantly, salience seems to depend on a confluence of factors, some of which are spatial, others temporal, and yet others semantic. It is unlikely that any one heuristic can capture the salience of an object. The framework within FrameSelect uses simple conditional logic to include new objects but the problem can be framed more generally as a probabilistic tracking and predicting problem similar to [6]. We can use a Kalman filter to elegantly incorporate multiple signals, and also significantly, incorporate uncertainty estimates into our mask updates.

If masks detected within each frame of a sequence are treated as a set of observations, augmented with other data such as depth (perhaps recovered from monocular vision estimates), relative motion, semantic importance, we can come up with an initial prediction of whether an object is salient, including a confidence estimate. Then, at each frame, we can update our prior beliefs and update our salience estimate and confidence based on changes in the observed data and our process model. If the confidence achieves a high enough level, we can act on our prediction and either add or remove the mask.

This work shows that it is possible to improve Unsupervised Video Object Segmentaton performance if certain assumptions are made about scene structure to help guide object mask salience, but at the cost of generality. Further work focused on incorporating multiple beliefs and a formal process model of salience posed as a Kalman filtering problem may further help improve performance.

Code for the project is available in a private repository at https://github.com/cornmander/cs231a-project-uvos.

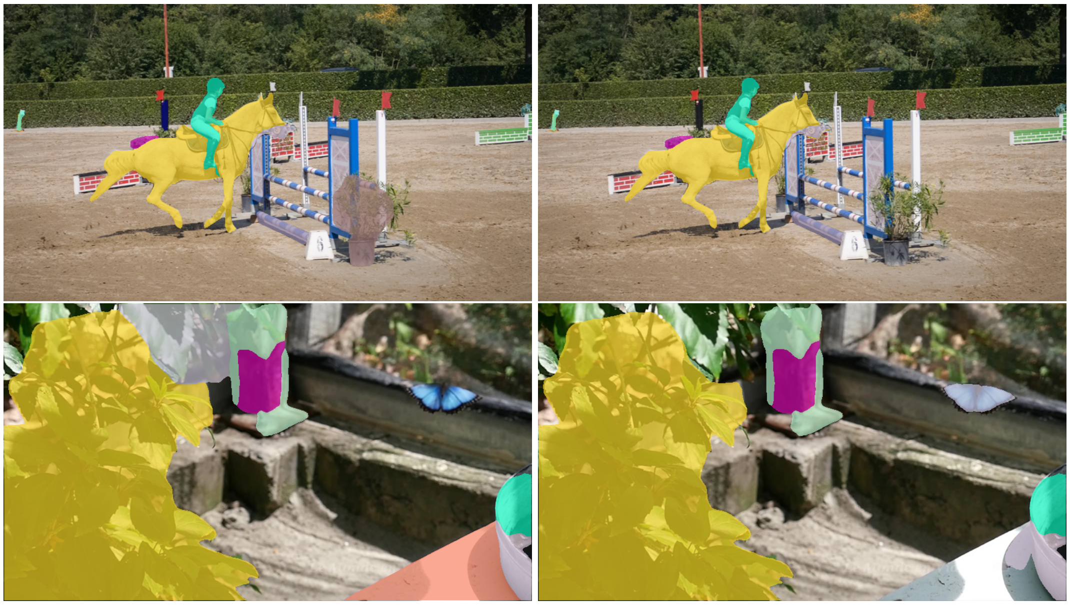

Figure 7. FrameSelect-Original vs. FrameSelect-S95 output. The non-salient object in the foreground is excluded in the FrameSelect-S95 output. In particular the butterfly in the bottom pair is completely ignored by FrameSelect-Original but tracked in FrameSelect-S95, which leads to a much better score on this sequence.

## References

[1] S. Caelles, J. Pont-Tuset, F. Perazzi, A. Montes, K.-K. Maninis, and L. Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv:1905.00737*, 2019. 1, 4

[2] J. Cheng, Y. Tsai, S. Wang, and M. Yang. Segflow: Joint learning for video object segmentation and optical flow. *CoRR*, abs/1709.06750, 2017. 2, 5

[3] S. Garg and V. Goel. Mask selection and propagation for unsupervised video object segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1680–1690, 2021. 1

[4] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1

[5] Y. Henon. pyimgsaliency. https://github.com/yhenon/pyimgsaliency, 2021. 3

[6] H. kuang Chiu, A. Prioletti, J. Li, and J. Bohg. Probabilistic 3d multi-object tracking for autonomous driving, 2020. 5

[7] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim. Video object segmentation using space-time memory networks. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019. 1

[8] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016. 1

[9] 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:1704.00675*, 2017. 4, 5

[10] S. K. S. Garg, V. Goel. Unsupervised video object segmentation using online mask selection and space-time memory networks. *The 2020 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*, 2020. 1

[11] L. Wixson. Detecting salient motion by accumulating directionally-consistent flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):774–780, 2000. 2

[12] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 3

[13] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2814–2821, 09 2014. 1, 2