CS 549 Scalable Querying
Data Augmentation for Video Object Segmentation and Tracking: a Survey

Austin Lally [1]

Abstract

Data augmentation has become a central strategy in computer vision models in recent years. It is possibly even more important in the video paradigm, where annotated data is that much more expensive and challenging to obtain. This paper reviews recent literature for data augmentation techniques, specifically as applied to video object tracking and video segmentation tasks. It identifies several promising approaches, catalogs which papers have used which augmentation methods, and explores a promising opportunity for future research.

1. Introduction

Object segmentation and tracking are two core problems in computer vision. In object segmentation, the task is to label each pixel of an input image to associate the pixel with a real-world object. Effectively, this means generating one or more masks indicating where objects exist in the image. Video object segmentation, then, refers to generating a mask for a particular object for every frame of a video. In object tracking, the goal is to not only locate one or more objects in each frame of a video (usually by producing a bounding box), but also to maintain continuity between frames so as to understand how each object moves over time.

While these are distinct problems, it is easy to see how they are related and how solving each can contribute to solving the other. Segmentation masks give an easy way to find bounding boxes for the object tracking problem, while bounding boxes from a tracking solution give a rough estimate of the segmentation problem and help demystify deformation and temporary occlusion. In fact, by solving one problem one must solve the other, at least partially [77].

Modern deep models for video object tracking and segmentation have huge numbers of parameters and require a lot of training data, but labeled video data is particularly expensive to create because of the large number of images (frames) that require annotations. When insufficient training data is available, models tend to overfit and fail to generalize well to test data. This paper explores the recent literature in search of data augmentation techniques to solve this data scarcity problem.

Several other survey-style papers [54; 77; 46; 44] proved invaluable in locating materials for this work.

In the next section, a wide variety of techniques are described. Basic techniques that are assumed to be well-understood are just listed along with a list of references where they have been used recently. More advanced techniques are described in detail. Section 3 proposes a future research possibility. Section 4 identifies some related works that fall outside the scope of this paper. Finally, closing thoughts are offered in section 5.

2. Techniques

There are quite a number of data augmentation techniques covered in the literature. They can be broadly categorized as heuristic or learned. Because segmentation and tracking in video are related so fundamentally to their underlying still image counterparts, all the standard data augmentation techniques for image problems apply. Hence, techniques can be further decomposed into video-specific techniques versus those more generally applicable to still images.

2.1. Heuristic Techniques for Images

Heuristic techniques are those that are hand-designed based on some assumption about the dataset distribution. For example, the intuition is that changing the brightness of an image doesn't change the location of a target object within it, so brightness-based augmentation is used for object detection data. Heuristic augmentation techniques for images fall into a few high-level categories which will be covered in the remainder of this section. See figure 1 for some select examples.

[1]School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR. Correspondence to: Austin Lally <lallya@oregonstate.edu>.

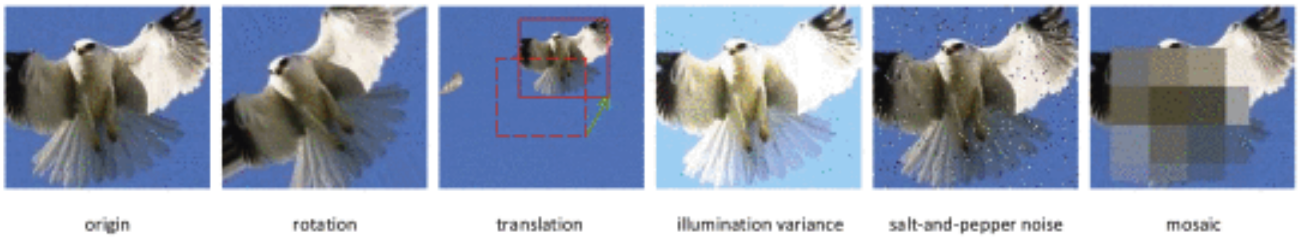

Figure 1. Heuristic augmentation examples (image credit [13]).

2.1.1. Geometric

Geometric data augmentations are those that change an image by somehow morphing its shape or orientation. This type of augmentation is used extensively through much of the literature. In no particular order, these include:

- Translation [12; 22; 62; 87; 16; 70; 15; 31; 48; 74; 5; 90; 10; 13; 23; 4; 24; 41; 25; 20; 60]
- Rotation [53; 12; 68; 22; 64; 76; 42; 88; 87; 6; 19; 39; 63; 79; 5; 13; 23; 4; 24; 41]
- Scaling [53; 12; 68; 62; 50; 64; 59; 83; 58; 88; 6; 19; 39; 61; 70; 15; 63; 66; 65; 74; 90; 47; 25; 75; 60]
- Flipping [43; 53; 12; 68; 50; 72; 59; 11; 40; 42; 88; 6; 19; 79; 38; 5; 47; 21; 23; 4; 24; 20]
- Shearing [12; 22; 19]
- Thin Plate Splines (TPS) [12; 64; 83; 39]

  Also known as elastic deformation, this method deforms the image by setting a number of control points and randomly shifting them within a specified size range.

Some papers [37; 81] indicate more generally that they use affine transforms, which means they are using some combination of geometric transformations.

2.1.2. Lighting

Lighting-based augmentations adjust images by changing their pixel values in a uniform way across the entire image.

- Grayscale [12; 59; 58; 88; 7; 90; 25]
- Gamma [43; 12; 59; 58]
- Brightness [12; 22; 35; 83; 16; 79; 74; 90]
- Contrast [12; 22; 35; 83; 74; 90]
- Saturation [12; 35; 83; 42; 74]
- Color [12; 22; 89; 63; 49; 74; 47; 23; 25]

2.1.3. Destructive

These methods work by removing some part of the image content in an irreversible way. This means adding noise or arbitrarily setting pixel values.

- Erasing [12; 72]

  This means setting pixel values to zero in randomly chosen rectangular regions within the image.
- Cropping [12; 64; 11; 76; 83; 39; 61; 31; 48; 18; 47; 41; 60]

  This is using a rectangular subregion of the image as input instead of the full original image.
- Blurring [12; 59; 58; 88; 87; 5; 90; 21; 13; 23; 4; 24; 20]

  These techniques blur the entire image with a Gaussian, median, or other blurring filter.
- Noise [13; 20]

  In these papers, the authors apply salt & pepper or Gaussian noise to the input images.
- Mosaic [13]

  Effectively a very low-resolution blur, this technique averages the pixel values across rectangular subregions of uniform size (see figure 1).

2.1.4. Dropout

Dropout is a different form of regularization. For each layer of a neural network, applying dropout means randomly choosing some percentage p of its weights for each training example and setting them to zero. Effectively, p of the layer's units are dropped out for that training example. Looking at this a different way, each example is passed through a randomly thinned sub-model, and the final trained model is an average over those sub-models. One intuition for how this improves performance is that it prevents co-adaptations, where a unit learns to compensate for mistakes of another unit. Because the set of neighboring units changes, a unit cannot learn to rely on the behavior of any other unit.

While this doesn't operate directly on the training data, and isn't generally viewed as a data augmentation technique, it does have a similar effect. Imagine that instead of applying dropout, some subset of the pixels of an image were set to zero. The outcome would be similar: the corresponding set of units would have no effect, because they would be multiplied by zero-valued pixels. The difference here is that zeros in an image will propagate through all layers of a network, but that dropout is applied on a per-layer basis. In the end, dropout behaves like a certain kind of data augmentation *in the latent feature space*.

Because of its generality, dropout is applicable to any kind of neural network and isn't limited to images or videos in particular. It is a very common regularization technique in the literature [12; 35; 34; 89; 5; 23; 4; 24].

2.1.5. Siamese Search-Exemplar Pairs

Following the 2015 addition to ImageNet of a video object detection dataset [51], a novel technique was proposed [3] for object tracking. They designed a fully convolutional Siamese network to solve a general similarity learning problem, and then applied it to the object tracking problem at evaluation time. While not a data augmentation technique in its own right, this formulation lends itself naturally to a particular style of data augmentation.

A Siamese network is an architecture designed for comparison problems. It applies an identical transformation to each of two inputs, and then combines their corresponding representations with another (possibly learned) transformation. As a straightforward example, the transformation applied to the inputs may be an embedding network, and the subsequent combinator might be a simple distance metric.

For the object detection problem, the authors designed a Siamese network that takes two images as inputs: a larger search image and a smaller exemplar image. By using a cross-correlation operator as the final comparison layer, they produced as output a two-dimensional grid of scores, where each score indicates the probability of the exemplar image appearing at the corresponding position in the search image. See figure 2 for an overview of the architecture.

They trained the model offline using both positive and negative image pairs from the ImageNet Video dataset. For each training pair, they selected two nearby frames from the same video. They used a large crop from one frame as the search image and a smaller crop from the other frame as the exemplar image. For positive examples, both crops were centered on the target object. The crop strategy for negative examples was not specified. The ground-truth labels for the score grid were set to +1 within a fixed radius of the object's true center, and -1 outside that radius. In this way, the network was trained to predict which region of the search

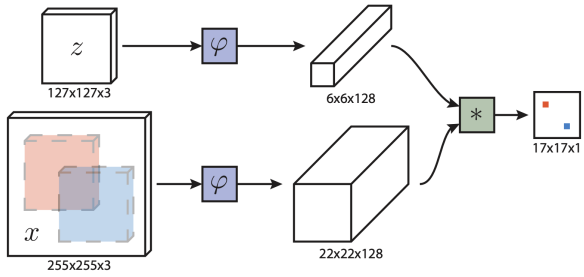

Figure 2. Siamese network architecture for object detection.

image was most similar to the target object.

At inference time, the embedding of the target object was computed once and then compared (at multiple scales) to search windows in each subsequent frame. Notably, no online fine-tuning was applied at test time, and the offline-trained network was directly tested against multiple benchmark datasets. Because no online fine-tuning was required, the model ran much more quickly than other contemporary models, and still showed very good results.

This network architecture is conducive to data augmentation because its training data is inherently generated from crops of larger images. By adjusting the cropping technique, many training examples can be created from each input image. This can be seen in many of the papers that built upon the architecture over the following years [62; 37; 86; 59; 88; 80; 87; 1; 16; 30; 78; 70; 90; 47; 13].

2.2. Learning-Based Techniques for Images

2.2.1. Smart Augmentation

In [36], the authors designed a type-agnostic learning-based augmentation method called Smart Augmentation. Given a downstream network that would benefit from data augmentation, they trained an augmentation network to provide useful augmentations for the downstream network. Specifically, their augmentation network learned to combine a set of inputs of a single class into a new example of that class. With the idea that the augmentation network should both improve the downstream network and generate examples that are similar to the existing training data, it was trained using a combination of two loss values. The loss from the downstream network was propagated back through the augmentation network, and the output of the augmentation network was compared to another training input from the same class for similarity. Figure 3 shows the network architecture and loss functions, and figure 4 shows an example of its output.

Through a number of experiments, they showed that Smart

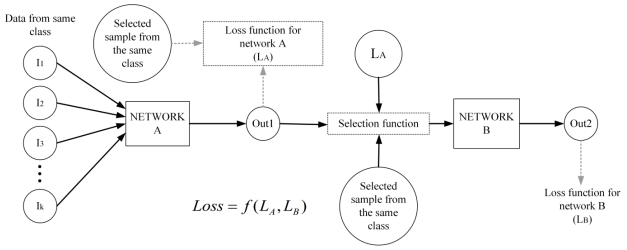

Figure 3. Smart Augmentation architecture.

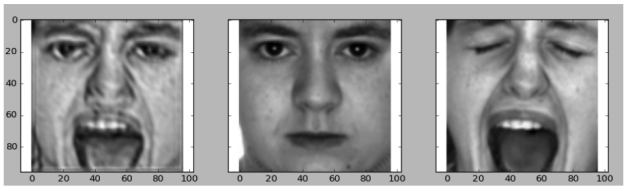

Figure 4. Sample image (left) generated by Smart Augmentation from two input images (center, right).

Augmentation could significantly improve the performance of downstream networks. They also showed that Smart Augmentation performed better than one particular selection of traditional augmentation techniques (flip, blur, and slight rotation). They showed that Smart Augmentation worked well in combination with other traditional augmentations, so it could be added to an existing training regimen without hurting performance.

More extensive experimentation would be needed to decide whether Smart Augmentation is more effective than traditional augmentations in general. There is also a trade-off, of course, in that the additional augmentation network takes additional computational resources to train. The authors did not quantify this change in training time in the paper. On the plus side, because of its general nature (it can be applied to any format of input data, so long as the data is separated into classes), this is a promising technique for extension to new problem types.

2.2.2. AutoAugment

AutoAugment [22] is a reinforcement learning technique that learns augmentation strategies for image classification tasks. The authors trained a controller network to learn data augmentation policies using a reward function based on how well the downstream network performed with the selected policy. Each learned augmentation policy was composed of five sub-policies. Each sub-policy was a sequence of two heuristic augmentation techniques, with corresponding probability and magnitude values. For example, one of the learned sub-policies in a successful ImageNet policy was Equalize with probability 0.4 and magnitude 4 followed by Rotate with probability 0.8 and magnitude 8. In the downstream network, each training example is augmented with one randomly selected sub-policy.

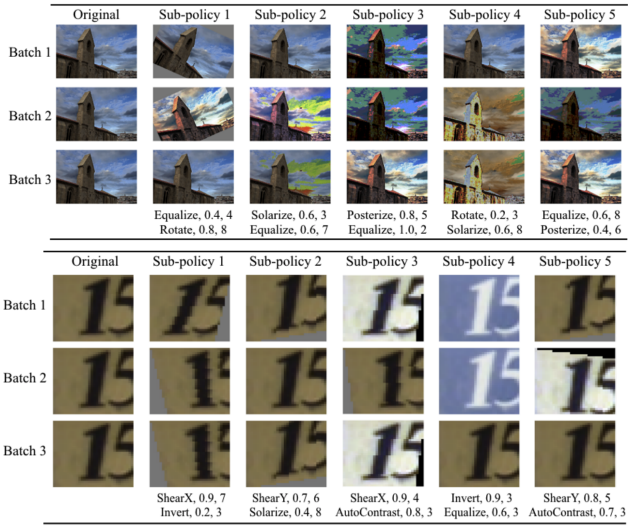

Figure 5. Examples of augmentation policies learned by AutoAugment for ImageNet and SVHN datasets.

Through their experiments, the authors showed that AutoAugment dramatically improved performance of baseline models on many image classification datasets. They also showed that the learned augmentations were somewhat transferable. For example, augmentation policies learned on ImageNet also performed well on several other datasets. However, the controller did learn to produce different strategies for different datasets, and directly training on the intended dataset produced even better results than transferring a policy learned for another dataset. As an example, the policies learned for ImageNet tended to include mostly color transformations while those learned for the house numbers dataset (SVHN) contained mostly geometric transformations (see figure 5).

This was a really strong paper and it set a new standard for learned data augmentation. The cost, as with any learned augmentation technique, is in the additional required training time. As with Smart Augmentation, however, this is a really interesting approach that could be adapted to any kind of downstream task. For other image-based tasks, like object detection or segmentation, this could be applied directly.

2.2.3. InstaBoost

InstaBoost [27] is another augmentation technique that has not been applied to video-based problems, but that might prove useful in that context. This has been shown to improve instance segmentation results by simply cut-and-pasting the

objects, along with their masks, to another position. When pasting the objects, the authors applied affine transformations including scale, rotation and translation following a probability distribution to place the objects in likely positions.

In their initial approach, the authors followed the simple intuition that images tend to be continuous and redundant at the pixel level. They posited that in a given image, the areas in the immediate neighborhood of an object would also have a high likelihood of containing the object. So they chose transformations randomly in a small neighborhood of the original position. This augmentation technique yielded significant improvements on instance segmentation benchmarks using state of the art models with no other adjustments.

To improve the performance further, they learned a more robust probability model from the training data itself. They expanded the feasible target region based on the assumption that objects are more likely to appear on semantically similar backgrounds than on different ones. They produced a heatmap of likely locations based on local appearance similarity of the background. Instead of choosing destinations in the local neighborhood of the original object position, they chose the new location according to the generated heatmap (see figure 6). This resulted in further improvements on the instance segmentation benchmarks.

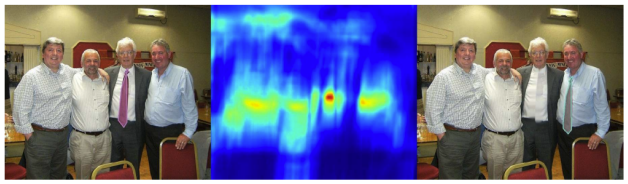

Figure 6. Input image (left) along with heatmap (center) and output image (right) generated by InstaBoost.

This is a really nice approach because it's simple and can be easily integrated into the training pipeline of any existing segmentation model. It could be used to improve augmentation / generation techniques for video data by guiding object placement toward plausible regions in the frame.

However, some choices were not explained very thoroughly. First, the authors divide the background region into three nested regions centered around the target object and weight the similarity metric differently for each region. While the intuition is mentioned, it's not clear why three was the correct number of regions or how the relative weights were chosen. It seems further experimentation is warranted here. Second, exhaustive comparison of all target regions is computationally expensive, so the authors downsample the images to a fixed size before heatmap generation. Again, the intuition is clear here, but some specifics as to how the target size was chosen and what other methods were considered would be helpful.

Overall, this is a nice paper despite wanting for some minor details.

2.3. Video-Specific Techniques

2.3.1. Semi-Supervision Using Graphical Models

One of the earliest approaches used specifically for data augmentation in video segmentation was [8], in which the authors designed a probabilistic graphical model for semi-supervised video segmentation. They modeled correspondences between patches of pixels over time as a tree structure, and trained a generative model to estimate patches of the current frame based on the previous frame. To combat bias, they also ran their model in the reverse direction (with the tree rooted in the last frame instead of the first) and averaged the results. Taking as input a video sequence and a single ground-truth segmentation mask for the first frame, they propagated that mask through all subsequent frames to effectively generate labeled masks for the entire video. The method was more computationally efficient and more effective than other competitive approaches at the time.

Because of the cost of manually labeling many frames of video data, most video datasets only label a small subset of keyframes for each video. In [9], the authors used the PGM method of [8] to automatically label additional video frames for three such datasets: CityScapes, CamVid, and CamVid-Instance. They then evaluated six segmentation models on the class segmentation datasets, and two segmentation models on the instance segmentation dataset. They compared results after training on just the original datasets with the results after training on the augmented data, and showed that training on the automatically-generated augmented data significantly improved both class and position accuracy.

This was effectively a transductive supervised learning approach, but it was an unusual example of actually generating and storing all of the augmented training data up front. It also stood out as the only quantitative measure of the effectiveness of data augmentation for video segmentation at the time.

2.3.2. Adversarial Learning: VITAL

One interesting new technique for handling inadequacies in training data came in 2018. In [55], the authors used a novel adversarial architecture in hopes of capturing rich appearance variations in positive training samples. While they argued that a traditional generative adversarial network (GAN) architecture is poorly suited for object tracking, they found a way to use a related approach.

In a standard GAN architecture, a discriminator network is typically used to improve the quality of the generator network, and the useful product of the effort is a generator that maps samples from one distribution to another (e.g., maps noise to realistic images). By contrast, in this work the valuable outcome was the improved discriminator—the generator was discarded.

The authors built upon a tracking-by-detection framework by inserting a mask-generating subnetwork between the feature extraction layer and the classification layer (see figure 7). The generated masks would hide features from the classifier, forcing it to learn to classify the existence of the target object on a subset of the available features. Meanwhile, the classifier loss was used to train the generator such that the generator learned to mask the most important features. In this way, the classifier learned to use only more robust long-term features instead of overfitting to the most discriminative features for individual frames.

Intuitively, imagine that the object to track is a person, and in the first few frames the classifier learns to recognize the face as the most important feature. If the person turns away from the camera, the classifier will struggle to identify the person in subsequent frames. In this adversarial masking approach, the mask generator learned to mask the face because it was the most important discriminative feature for the classifier. In turn, the classifier learned to recognize the person based on other features that were less likely to change from frame to frame. From another perspective, this approach was a lot like the commonly-used random erasing data augmentation, except that the erasing was not random at all (or rather, it became less random through training).

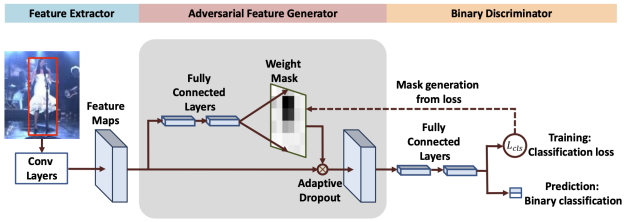

Figure 7. Network architecture for VITAL [55].

One major drawback to this approach is the computational cost of introducing the mask-generating subnetwork. They claim only 1.5 FPS. It would be interesting to see a comparison of runtime between this network and a version without the mask generator to understand how much overhead it creates.

2.3.3. Lucid Data Dreaming

Lucid Data Dreaming for Video Object Segmentation [33] is a particularly interesting paper, and really feels more like two papers in one. First, the authors designed LucidTracker, a convolutional video object segmentation and tracking network. Additionally, they devised Lucid Data Dreaming, a method for synthesizing video data from a single image and its corresponding mask.

For the LucidTracker network, they modeled the problem as predicting an object mask for the current frame given a five channel input: the RGB channels for the current frame plus the mask and optical flow information from the previous frame. They also extended the approach to support multiple object tracking by incorporating additional input channels - one mask channel per object. They discussed several training modalities, from a training-free, hand-designed approach all the way down to fine-tuning for each video. In general, though, they found that their model benefited from training on labeled video data, which is expensive and hard to come by.

Lucid Data Dreaming was their solution to the data problem. Given a single input image and corresponding object mask, they generated a pair of plausible adjacent video frames with known ground-truth masks and optical flow. They did this by cutting the target object out of the input image using the given mask, infilling the background, transforming both foreground and background images, and then pasting the foreground object into a new position in the image. The new object position was uniformly sampled for the first frame, but then the difference between the two frames was kept small. The transformations included illumination changes, affine and non-rigid transformations of the foreground object, and affine transformations of the background to simulate camera motion. Figure 8 shows the high-level concept and some examples of generated frames.

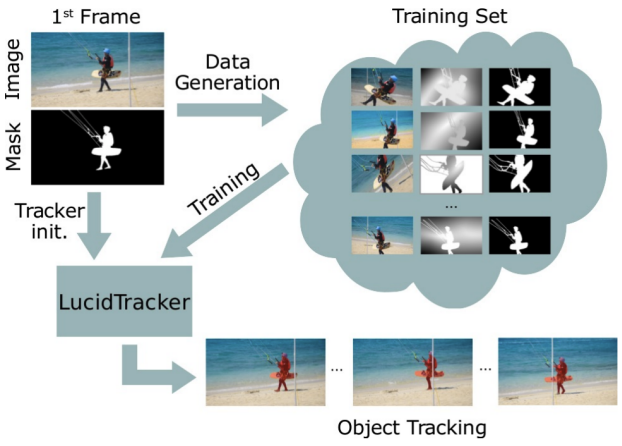

Figure 8. LucidTracker / Lucid Data Dreaming concept.

The authors tested on three video object segmentation datasets in which ground-truth masks were provided for each frame. For each video, they trained on only the first

frame along with 2,500 pairs of generated plausible future video frames. They achieved better results than both flow propagation methods and other deep models, and they used significantly less training data than the competing deep learning models.

While their tracker model was very successful, it's the data generation portion that proved even more interesting. It was used to generate training data in a number of subsequent papers [43; 68; 2; 57; 69; 52]. There are two main drawbacks to the paper, however. First, the data generation approach is computationally expensive. Including fine-tuning first for the dataset and then for each video, their approach took about 3.5 hours per video. While fine-tuning for each video is not required for their model to work, it does improve the results. A useful experiment would be to try augmenting the dataset with just one frame pair, or some small number of pairs based on the amount of time taken to train on them. Can this technique be applied to good effect in a (near) real-time context? Second, it would have been good to see a comparison between their data generation method and other data augmentation techniques. Would a simpler approach such as random transformations work just as well? In any case, this one of very few promising methods for generating video training data in the literature.

3. Open Problem

3.1. Learning-Based Augmentation for Video Problems

There has been very little work on learned data augmentations for video data. However, learning-based approaches have shown some of the best results for other computer vision tasks. AutoAugment [22] in particular is appealing because of its generality. Since the same reinforcement learning-based technique can be used to learn any sort of policy, there is plenty of opportunity to apply it in the video domain. The remainder of this section will discuss a proposed approach. See figure 9 for an overview of the proposed architecture.

3.1.1. Object Tracker

The core network to train is a Siamese object tracking-by-detection framework, as in [3]. Because these have been well-studied, there is a good deal of past work against which to compare results.

3.1.2. Augmentations

In order to keep the computation tractable while incorporating the most interesting techniques in recent literature, augmentation policies will be composed of three augmentation techniques.

- VITAL [55]

  For consistency with the other two augmentations, one major modification is made to this adversarial mask learning technique. Instead of learning masks in feature space, the subnetwork is trained to generate masks in the input image space, before crops are chosen for the Siamese network. The magnitude parameter, ranging from 0 to 0.5, controls how much of the input image is masked.

- Lucid Data Dreaming [33]

  Following this approach, the target object is cropped out of the input image, transformed, and replaced. The magnitude parameter, ranging between 0 and 1, controls the degree of scaling, rotation, and TPS deformation applied.

- Smart Augmentation [36]

  This technique learns to interpolate between multiple input images to form a new in-distribution image. In this application, it will combine two upcoming video frames to form a new plausible video frame. The magnitude parameter, ranging between 2 and 10, controls the maximum distance to the farthest-future frame considered. For example, with magnitude 2 the next frame and subsequent frame are always combined via Smart Augmentation, but with magnitude 10 the next frame is combined with a frame randomly chosen from among the next 10.

3.1.3. Policy Design

As in AutoAugment [22], each data augmentation policy is a set of five sub-policies and each sub-policy is a sequence of two of the above augmentation techniques. For each sub-policy, the chosen strategy is parameterized by two values: the probability that it is applied, and the magnitude of the operation if it is applied.

3.1.4. Controller Network

The controller network follows the architecture defined in AutoAugment [22]. Specifically, it is a single-layer LSTM with 100 hidden units and a 30-dimensional linear output layer with softmax predictions.

3.1.5. Training

The training approach also follows AutoAugment [22]. At each training iteration, the controller network produces a policy in the form of a 30-dimensional vector (operation type, probability, and magnitude for each of two steps for each of five policies). The controller is trained using proximal policy optimization (PPO), and its reward signal depends on how well the child network (the Siamese object tracker) generalizes after training with the predicted policy.

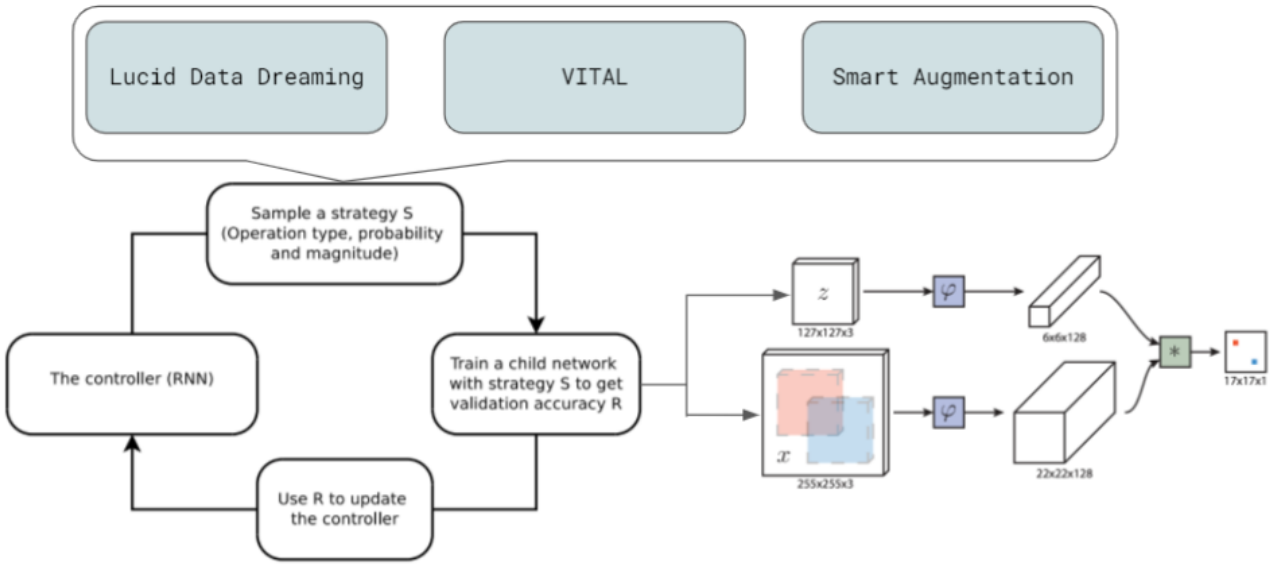

Figure 9. Proposed learned video data augmentation architecture.

To generate the reward signal, the predicted policy is used to train the child network on the training data set. Then the child network is evaluated on the validation set, and the resulting accuracy is used as the reward value. The reward value scales the gradient of the controller network before backpropagation, so that the controller network assigns high probabilities for successful child networks and low probabilities for poor ones.

4. Related Areas

This section briefly mentions some interesting related areas. Although outside the scope of this review, the papers listed here may pique the interest of the reader.

First, there are three other problem definitions related to video object segmentation and tracking that have appeared more recently. Multi-Object Tracking and Segmentation [56] and Video Instance Segmentation [73] extend the video segmentation problem to segmenting multiple distinct object instances, and Multi-Object Panoptic Tracking [32] goes one step further and requires segmentation of background objects as well.

Next, several papers were encountered that design adversarial attacks for visual trackers. This kind of technique could be used as regularization or data augmentation as in [55], but it was outside the scope of this work to review them in depth. Some of the standout examples of adversarial attacks are [29; 71; 17].

There was also a line of work specifically around tracking in blurred video [28; 45; 67]. Augmentation could of course help generate blurred video from non-blurred video, and in general techniques that are successful on blurred video are also successful on ordinary video.

TracKlinic [26] and Learning Multi-Object Tracking and Segmentation from Automatic Annotations [50] both seek, in one way or another, to evaluate what kinds of augmentations could be beneficial for what kinds of data. This could be useful for designing distribution-invariant transformation groups as described in A Group-Theoretic Framework for Data Augmentation [14].

There is also some work on tracking with data other than just video. One author, for example, has produced a few works [85; 82; 84] on adding infrared imagery to video data. This is an interesting line of thinking. If the real-world application is to track objects in space, it is probably worthwhile to seek other types of data to aid in the problem.

5. Conclusion

Data augmentation, while central to much of the recent computer vision work, is relatively underrepresented in the video segmentation and tracking literature. While many authors are using traditional heuristic augmentations, it appears that they are doing so somewhat arbitrarily. The most likely explanation is actually that authors are applying the easiest augmentations at their disposal, based on what is available in their deep learning framework of choice.

Novel techniques for video data are rare and hard to find, though the few that have appeared in recent years have been very successful. Further exploration of these techniques is

worthwhile. Since years have passed since the introduction of some of the important video-specific approaches, it would be very interesting to see them applied to new problems on new hardware.

This paper covered a wide range of augmentation techniques ranging from simple to clever. It gave an indication of how often the various heuristic techniques are applied, and gave a thorough summary of a handful of the more specific methods. It also proposed a way to combine reinforcement learning with some of the more interesting video augmentation tools to learn which ones are best-suited to which datasets. Finally, it included some pointers to related research areas. It is hopefully a useful summary for newcomers to the field or for those looking for potential open problem areas.

References

[1] Mohamed H Abdelpakey and Mohamed S Shehata. Domainsiam: Domain-aware siamese network for visual object tracking. In *International Symposium on Visual Computing*, pages 45–58. Springer, 2019.

[2] Linchao Bao, Baoyuan Wu, and Wei 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.

[3] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking, 2016.

[4] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning Discriminative Model Prediction for Tracking. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6181–6190, Seoul, Korea (South), October 2019. IEEE.

[5] Goutam Bhat, Joakim Johnander, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Unveiling the power of deep tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 483–498, 2018.

[6] Goutam Bhat, Felix Järemo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. Learning what to learn for video object segmentation. In *European Conference on Computer Vision*, pages 777–794. Springer, 2020.

[7] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2544–2550. IEEE, 2010.

[8] Ignas Budvytis, Vijay Badrinarayanan, and Roberto Cipolla. Semi-supervised video segmentation using tree structured graphical models. In *CVPR 2011*, pages 2257–2264, 2011.

[9] Ignas Budvytis, Patrick Sauer, Thomas Roddick, Kesar Breen, and Roberto Cipolla. Large scale labelled video data augmentation for semantic segmentation in driving scenarios. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 230–237, 2017.

[10] Elena Burceanu and Marius Leordeanu. Learning a robust society of tracking parts using co-occurrence constraints. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[11] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017.

[12] Ángela Casado-García, César Domínguez, Manuel García-Domínguez, Jónathan Heras, Adrián Inés, Eloy Mata, and Vico Pascual. Clodsa: a tool for augmentation in classification, localization, detection, semantic segmentation and instance segmentation tasks. *BMC bioinformatics*, 20(1):1–14, 2019.

[13] Kai Chen and Wenbing Tao. Once for all: a two-flow convolutional neural network for visual tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(12):3377–3386, 2017.

[14] Shuxiao Chen, Edgar Dobriban, and Jane H Lee. A group-theoretic framework for data augmentation, 2020.

[15] Xi Chen, Zuoxin Li, Ye Yuan, Gang Yu, Jianxin Shen, and Donglian Qi. State-aware tracker for real-time video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9384–9393, 2020.

[16] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. *arXiv preprint arXiv:2103.15436*, 2021.

[17] Xuesong Chen, Xiyu Yan, Feng Zheng, Yong Jiang, Shu-Tao Xia, Yong Zhao, and Rongrong Ji. One-shot adversarial attacks on visual tracking with dual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10176–10185, 2020.

[18] Zhizhen Chi, Hongyang Li, Huchuan Lu, and Ming-Hsuan Yang. Dual deep network for visual tracking. *IEEE Transactions on Image Processing*, 26(4):2005–2015, 2017.

[19] Suhwan Cho, MyeongAh Cho, Tae-young Chung, Heansung Lee, and Sangyoun Lee. Crvos: Clue refining network for video object segmentation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 2301–2305. IEEE, 2020.

[20] Janghoon Choi, Junseok Kwon, and Kyoung Mu Lee. Deep Meta Learning for Real-Time Target-Aware Visual Tracking. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 911–920, Seoul, Korea (South), October 2019. IEEE.

[21] Jongwon Choi, Hyung Jin Chang, Tobias Fischer, Sangdoo Yun, Kyuewang Lee, Jiyeoup Jeong, Yiannis Demiris, and Jin Young Choi. Context-aware deep feature compression for high-speed visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 479–488, 2018.

[22] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019.

[23] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: Accurate Tracking by Overlap Maximization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4655–4664, Long Beach, CA, USA, June 2019. IEEE.

[24] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic Regression for Visual Tracking. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7181–7190, Seattle, WA, USA, June 2020. IEEE.

[25] Fei Du, Peng Liu, Wei Zhao, and Xianglong Tang. Correlation-Guided Attention for Corner Detection Based Visual Tracking. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6835–6844, Seattle, WA, USA, June 2020. IEEE.

[26] Heng Fan, Fan Yang, Peng Chu, Yuewei Lin, Lin Yuan, and Haibin Ling. Tracklinic: Diagnosis of challenge factors in visual tracking. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 970–979, 2021.

[27] Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 682–691, 2019.

[28] Qing Guo, Wei Feng, Ruijun Gao, Yang Liu, and Song Wang. Exploring the effects of blur and deblurring to visual object tracking. *IEEE Transactions on Image Processing*, 30:1812–1824, 2021.

[29] Qing Guo, Xiaofei Xie, Felix Juefei-Xu, Lei Ma, Zhongguo Li, Wanli Xue, Wei Feng, and Yang Liu. Spark: Spatial-aware online incremental attack against visual tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 2. Springer, 2020.

[30] Guang Han, Hua Du, Jixin Liu, Ning Sun, and Xiaofei Li. Fully conventional anchor-free siamese networks for object tracking. *IEEE Access*, 7:123934–123943, 2019.

[31] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *European conference on computer vision*, pages 749–765. Springer, 2016.

[32] Juana Valeria Hurtado, Rohit Mohan, Wolfram Burgard, and Abhinav Valada. Mopt: Multi-object panoptic tracking. *arXiv preprint arXiv:2004.08189*, 2020.

[33] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for video object segmentation, 2019.

[34] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6479–6488, 2020.

[35] Zihang Lai and Weidi Xie. Self-supervised learning for video correspondence flow. *arXiv preprint arXiv:1905.00875*, 2019.

[36] Joseph Lemley, Shabab Bazrafkan, and Peter Corcoran. Smart augmentation learning an optimal data augmentation strategy. *Ieee Access*, 5:5858–5869, 2017.

[37] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8971–8980, 2018.

[38] Hanxi Li, Yi Li, and Fatih Porikli. Deeptrack: Learning discriminative feature representations online for

robust visual tracking. *IEEE Transactions on Image Processing*, 25(4):1834–1848, 2015.

[39] Yu Li, Zhuoran Shen, and Ying Shan. Fast video object segmentation using the global context module. In *European Conference on Computer Vision*, pages 735–750. Springer, 2020.

[40] Huaijia Lin, Xiaojuan Qi, and Jiaya Jia. Agss-vos: Attention guided single-shot video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3949–3957, 2019.

[41] Xiankai Lu, Bingbing Ni, Chao Ma, and Xiaokang Yang. Learning transform-aware attentive network for object tracking. *Neurocomputing*, 349:133–144, July 2019.

[42] Xinkai Lu, Wenguan Wang, Martin Danelljan, Tianfei Zhou, Jianbing Shen, and Luc Van Gool. Video object segmentation with episodic graph memory networks. *arXiv preprint arXiv:2007.07020*, 2020.

[43] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation, 2018.

[44] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. Multiple object tracking: A literature review. *Artificial Intelligence*, page 103448, 2020.

[45] Bo Ma, Lianghua Huang, Jianbing Shen, Ling Shao, Ming-Hsuan Yang, and Fatih Porikli. Visual tracking under motion blur. *IEEE Transactions on Image Processing*, 25(12):5867–5876, 2016.

[46] Seyed Mojtaba Marvasti-Zadeh, Li Cheng, Hossein Ghanei-Yakhdan, and Shohreh Kasaei. Deep learning for visual tracking: A comprehensive survey. *IEEE Transactions on Intelligent Transportation Systems*, page 1–26, 2021.

[47] Henrique Morimitsu. Multiple context features in siamese networks for visual object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[48] Hyeonseob Nam and Bohyung 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.

[49] Bo Pang, Yizhuo Li, Yifan Zhang, Muchen Li, and Cewu Lu. Tubetk: Adopting tubes to track multi-object in a one-step training model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6308–6318, 2020.

[50] Lorenzo Porzi, Markus Hofinger, Idoia Ruiz, Joan Serrat, Samuel Rota Bulo, and Peter Kontschieder. Learning multi-object tracking and segmentation from automatic annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6846–6855, 2020.

[51] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015.

[52] Amirreza Shaban, Alrik Firl, Ahmad Humayun, Jialin Yuan, Xinyao Wang, Peng Lei, Nikhil Dhanda, Byron Boots, James M Rehg, and Fuxin Li. Multiple-instance video segmentation with sequence-specific object proposals. In *CVPR Workshop*, volume 1, 2017.

[53] Gilad Sharir, Eddie Smolyansky, and Itamar Friedman. Video object segmentation using tracked object proposals, 2017.

[54] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

[55] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson WH Lau, and Ming-Hsuan Yang. Vital: Visual tracking via adversarial learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8990–8999, 2018.

[56] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7942–7951, 2019.

[57] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for the 2017 davis challenge on video object segmentation. In *The 2017 DAVIS Challenge on Video Object Segmentation-CVPR Workshops*, volume 5, 2017.

[58] Paul Voigtlaender, Jonathon Luiten, and Bastian Leibe. Boltvos: Box-level tracking for video object segmentation. *arXiv preprint arXiv:1904.04552*, 2019.

[59] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian 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.

[60] Guangting Wang, Chong Luo, Xiaoyan Sun, Zhiwei Xiong, and Wenjun Zeng. Tracking by Instance Detection: A Meta-Learning Approach. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6287–6296, Seattle, WA, USA, June 2020. IEEE.

[61] 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.

[62] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip H. S. Torr. Fast online object tracking and segmentation: A unifying approach, 2019.

[63] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *arXiv preprint arXiv:1909.12605*, 2(3):4, 2019.

[64] Ziqin Wang, Jun Xu, Li Liu, Fan Zhu, and Ling Shao. Ranet: Ranking attention network for fast video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3978–3987, 2019.

[65] Mark Weber, Jun Xie, Maxwell Collins, Yukun Zhu, Paul Voigtlaender, Hartwig Adam, Bradley Green, Andreas Geiger, Bastian Leibe, Daniel Cremers, et al. Step: Segmenting and tracking every pixel. *arXiv preprint arXiv:2102.11859*, 2021.

[66] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris M Kitani. Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6499–6508, 2020.

[67] Yi Wu, Haibin Ling, Jingyi Yu, Feng Li, Xue Mei, and Erkang Cheng. Blurred target tracking by blur-driven tracker. In *2011 International Conference on Computer Vision*, pages 1100–1107. IEEE, 2011.

[68] Kai Xu, Longyin Wen, Guorong Li, Liefeng Bo, and Qingming Huang. Spatiotemporal cnn for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1379–1388, 2019.

[69] Shuangjie Xu, Daizong Liu, Linchao Bao, Wei Liu, and Pan Zhou. Mhp-vos: Multiple hypotheses propagation for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 314–323, 2019.

[70] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12549–12556, 2020.

[71] Bin Yan, Dong Wang, Huchuan Lu, and Xiaoyun Yang. Cooling-shrinking attack: Blinding the tracker with imperceptible noises. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 990–999, 2020.

[72] Bin Yan, Haojie Zhao, Dong Wang, Huchuan Lu, and Xiaoyun Yang. 'skimming-perusal'tracking: A framework for real-time and robust long-term tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2385–2393, 2019.

[73] 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.

[74] Tianyu Yang and Antoni B Chan. Recurrent filter learning for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2010–2019, 2017.

[75] Tianyu Yang, Pengfei Xu, Runbo Hu, Hua Chai, and Antoni B. Chan. ROAM: Recurrently Optimizing Tracking Model. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6717–6726, Seattle, WA, USA, June 2020. IEEE.

[76] Zhao Yang, Qiang Wang, Luca Bertinetto, Weiming Hu, Song Bai, and Philip HS Torr. Anchor diffusion for unsupervised video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 931–940, 2019.

[77] Rui Yao, Guosheng Lin, Shixiong Xia, Jiaqi Zhao, and Yong Zhou. Video object segmentation and tracking: A survey, 2019.

[78] Jihun Yoon, Seungbum Hong, Sanha Jeong, and Min-Kook Choi. Semi-supervised object detection with sparsely annotated dataset. *arXiv preprint arXiv:2006.11692*, 2020.

[79] Kwangjin Yoon, Jeonghwan Gwak, Young-Min Song, Young-Chul Yoon, and Moon-Gu Jeon. Oneshotda: Online multi-object tracker with one-shot-learning-based data association. *IEEE Access*, 8:38060–38072, 2020.

[80] Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R Scott. Deformable siamese attention networks for visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6728–6737, 2020.

[81] Xiaohui Zeng, Renjie Liao, Li Gu, Yuwen Xiong, Sanja Fidler, and Raquel Urtasun. Dmm-net: Differentiable mask-matching network for video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3929–3938, 2019.

[82] Pengyu Zhang, Jie Zhao, Chunjuan Bo, Dong Wang, Huchuan Lu, and Xiaoyun Yang. Jointly modeling motion and appearance cues for robust rgb-t tracking. *IEEE Transactions on Image Processing*, 30:3335–3347, 2021.

[83] Xiaofan Zhang, Haoming Lu, Cong Hao, Jiachen Li, Bowen Cheng, Yuhong Li, Kyle Rupnow, Jinjun Xiong, Thomas Huang, Honghui Shi, et al. Skynet: a hardware-efficient method for object detection and tracking on embedded systems. *arXiv preprint arXiv:1909.09709*, 2019.

[84] Xingchen Zhang, Ping Ye, Henry Leung, Ke Gong, and Gang Xiao. Object fusion tracking based on visible and infrared images: A comprehensive review. *Information Fusion*, 63:166–187, 2020.

[85] Xingchen Zhang, Ping Ye, Shengyun Peng, Jun Liu, Ke Gong, and Gang Xiao. Siamft: An rgb-infrared fusion tracking method via fully convolutional siamese networks. *IEEE Access*, 7:122122–122133, 2019.

[86] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. *arXiv preprint arXiv:2006.10721*, 2020.

[87] Jinghao Zhou, Peng Wang, and Haoyang Sun. Discriminative and robust online learning for siamese visual tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13017–13024, 2020.

[88] Wenzhang Zhou, Longyin Wen, Libo Zhang, Dawei Du, Tiejian Luo, and Yanjun Wu. Siamman: Siamese motion-aware network for visual tracking. *arXiv preprint arXiv:1912.05515*, 2019.

[89] Fangrui Zhu, Li Zhang, Yanwei Fu, Guodong Guo, and Weidi Xie. Self-supervised video object segmentation. *arXiv preprint arXiv:2006.12480*, 2020.

[90] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 101–117, 2018.