SALISA: Saliency-based Input Sampling for Efficient Video Object Detection

Babak Ehteshani Bejnordi, Amirhossein Habibian, Fatih Porikli, and Amir Ghodrati

Qualcomm AI Research*
{behtesha,ahabibia,fporikli,ghodrati}@qti.qualcomm.com

Abstract. High-resolution images are widely adopted for high-performance object detection in videos. However, processing high-resolution inputs comes with high computation costs, and naive down-sampling of the input to reduce the computation costs quickly degrades the detection performance. In this paper, we propose SALISA, a novel non-uniform SALiency-based Input SAmpling technique for video object detection that allows for heavy down-sampling of unimportant background regions while preserving the fine-grained details of a high-resolution image. The resulting image is spatially smaller, leading to reduced computational costs while enabling a performance comparable to a high-resolution input. To achieve this, we propose a differentiable resampling module based on a thin plate spline spatial transformer network (TPS-STN). This module is regularized by a novel loss to provide an explicit supervision signal to learn to "magnify" salient regions. We report state-of-the-art results in the low compute regime on the ImageNet-VID and UA-DETRAC video object detection datasets. We demonstrate that on both datasets, the mAP of an EfficientDet-D1 (EfficientDet-D2) gets on par with EfficientDet-D2 (EfficientDet-D3) at a much lower computational cost. We also show that SALISA significantly improves the detection of small objects. In particular, SALISA with an EfficientDet-D1 detector improves the detection of small objects by 77%, and remarkably also outperforms EfficientDet-D3 baseline.

Keywords: Video object detection, Saliency, Resampling, Efficient Object Detection, Spatial Transformer

1 Introduction

The rise in the quality of image capturing devices such as 4K cameras has enabled AI solutions to discover the most detailed video contents and, therefore, allowed them to be widely adopted for high-performance object detection in videos. However, the increased recognition performance resulting from this higher resolution signal comes with increased computational costs. This limits the application of state-of-the-art video object detectors on resource-constrained devices. As such, designing efficient object detection methods for processing high-resolution video streams becomes crucial for a wide range of real-world applications such as autonomous driving, augmented reality, and video surveillance.

* Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc

To enable efficient video object detection, a large body of works has been focusing on reducing feature computation on visually-similar adjacent video frames [22,23,4] [9,17,41,40]. This is achieved by interleaving heavy and light feature extractors [17], limiting the computation to a local window [9,4], or extrapolating features from a key frame to subsequent frames using a light optical flow predictor [41,40]. However, these approaches either suffer from feature misalignment resulting from two different feature extractors, or inefficiency in dealing with frequent global scene changes.

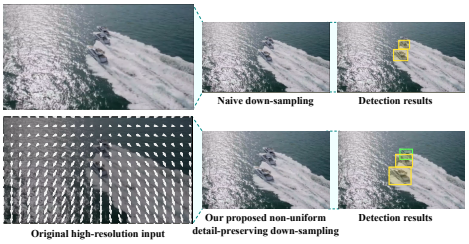

Fig. 1: An illustration of a non-uniform detail-preserving downsampling of input by SALISA leading to improved detection results.

An alternative approach to efficient object detection is to focus on designing lightweight yet highly accurate architectures such as EfficientDet [33]. Recent astounding advances in developing such models have deemed some of the above efficient approaches no longer applicable. For instance, flow-based feature extrapolation might no longer be a proper substitute for existing efficient feature extractors [33,32], as the cost of flow computation is no longer negligible. To be more specific, EfficientDet-D0 [33] costs only 2.5 GFLOPs per frame, while estimating flow by FlowNet-Inception [5] alone costs 1.8 GFLOPs, translating to 72% of the backbone itself [23]. However, such efficient architectures may still be expensive when applied to high-resolution video frames. On the other hand, naive down-sampling of the input to reduce the computation costs quickly degrades the performance [33,42]. For example, the performance of EfficientDet-D6 on COCO [15] degrades from 52.6% to 47.6% when the input is down-sampled by a factor of two [33].

In this work, we propose SALISA, a novel non-uniform input sampling technique that retains the fine-grained details of a high-resolution image while allowing for heavy down-sampling of unimportant background regions (see Figure 1). The resulting detail-preserved image is spatially smaller, leading to reduced computational cost but at the same time enabling a performance comparable to a high-resolution input. Given a sequence of video frames, we first apply a high performing detection model on a high-resolution input at $T = 1$ (without resampling). We then generate a saliency map from the detection output to guide the detailed-preserving resampling for the next high-resolution frame. This is achieved via a resampling module that applies a thin plate spline (TPS) [6] transformation to warp the high-resolution input to a down-scaled, detail-preserved one. The resulting resampled frame is then fed to the detector, which consequently has an easier job detecting objects at a lower computational cost.

Our resampling module is based on a thin plate spline spatial transformer network (TPS-STN) [10]. TPS-STN was originally proposed for image classification and used the task loss to train the parameters of STN for digit recognition in MNIST and SVHN. However, adapting this training scheme to object detection in natural images is nontrivial as STN cannot learn to "magnify" salient regions without an explicit supervision signal. To address this, we propose a loss term that imposes STN to mimic a content-aware

up-sampler. In particular, we use a weighted ℓ_2-loss between the sampling grid generated by our TPS-STN and the non-parametric attention-based sampler [38] designed for preserving details. Unlike the non-parametric approaches such as attention-based sampler [38] or classical seam carving techniques [29,1], our regularized sampling module is fully differentiable, computationally inexpensive, and generates distortion-free outputs.

Our contributions are as follows:

- We propose a novel efficient framework for video object detection. Using a saliency map obtained from a previous frame, we perform a non-uniform detail-preserving down-sampling of the current frame, enabling an accurate prediction at a lower computational cost.
- To perform the resampling, we develop a fully differentiable resampling module based on a thin plate spline spatial transformer network. We propose a new regularization technique that enables a more effective transformation of the input.
- We report state-of-the-art results in the low compute regime on the ImageNet-VID and UA-DETRAC video object detection datasets. In particular, we demonstrate that on both datasets, the mAP of an EfficientDet-D1 (EfficientDet-D2) gets on par with EfficientDet-D2 (EfficientDet-D3) at a much lower computational cost.

2 Related Work

Efficient video Object detection A straightforward approach to efficient video object detection is to apply existing efficient object detectors [27,18,2,3,42,33] on a per-frame basis. However, such an approach does not take the temporal redundancy into account and therefore is computationally sub-optimal for video object detection. In this paper, we specifically use the state-of-the-art cost-effective detection model EfficientDet [33] as our baseline and further extend it for video object detection.

Several methods are proposed to leverage temporal coherency between adjacent frames by tracking previous object detections to skip current detection [19,22], using template matching to learn patchwise correlation features in adjacent frames [23], limiting the feature computation by processing only a small sub-window of the frames [4,9], using heavy and light networks in an interleaving manner [17], or efficiently propagating features via a light FlowNet [41,40]. However, these methods might suffer from tracking errors, misalignments between features, or finding a suitable sub-window. Moreover, with existing efficient backbones [32], one may find out flow-based techniques no longer yield significant speed-ups, as the cost of flow computation is not negligible. As an alternative, we propose to resample the frame such that it retains the fine-grained details while allowing for heavy downsampling of background areas. The resulting image is spatially smaller, leading to a reduction in computation cost while enabling a performance comparable to a high-resolution input.

Adaptive Spatial Sampling One of the major challenges in object detection is to represent and detect fine-grained details in high-resolution images efficiently. One way to tackle this problem is to use hierarchical representations. [36,12,30,8] introduce hierarchical methods to refine the processing of a high-resolution image by adaptively

zooming into their proper scales. However, such a hierarchical processing approach makes these methods less suitable for real-time applications.

An alternative approach is to adaptively transform the input such that important fine-grained details are better preserved [10,26,38,7]. The pioneering work of Spatial Transformer Networks (STN) [10] proposes a differentiable module that enables a generic class of input transformations such as affine, projective, and thin plate spline transformations. While STN works well for MNIST and SVHN datasets, without explicit supervision, it has a hard job of learning effective transformations for complex recognition tasks. Learning-to-zoom [26] uses saliency maps generated by a CNN as guidance to performing a nonuniform sampling that magnifies small details. However, this method causes substantial deformation in the vicinity of the magnified regions, which is particularly harmful when objects overlap or positioned next to each other. Trilinear attention sampling network [38] aims to learn subtle feature representations from hundreds of part proposals for fine-grained image recognition. This technique overcomes the undesirable deformations observed in [26]. However, it is computationally more expensive, non-differentiable, and may still generate undesirable deformations in the background or lower saliency regions. Our method is based on a thin plate spline STN and employs [38] to supervise the STN, allowing it to work on complex datasets while largely eliminating the undesirable distortions caused by [38].

The adaptive spatial sampling techniques discussed above were primarily designed for image classification tasks. However, optimizing these techniques for downstream tasks such as object detection and semantic segmentation is more challenging. In particular, an undesirable deformation on a non-salient region is unlikely to harm the output prediction of a classification network. At the same time, it can deteriorate the performance of object detection or semantic segmentation model. Jin et al. [11] have proposed to use the learning-to-zoom approach [26] for adaptive downsampling of the input for semantic segmentation. To discourage the network from a naive sampling of easy-to-segment regions like background, the authors add an edge loss introduced in [24]. Recently, [34] has proposed a magnification layer based on learning-to-zoom [26] to resample pixels such that background pixels make room for salient pixels of interest. While the major focus of [26] is on improving object detection accuracy on small objects, we concentrate on increasing efficiency and at the same time improving the performance.

3 SALISA

Given a set of high-resolution video frames and their labels $\{\mathbf{f}_i, \mathbf{y}_i\}_{i=1}^{N}$, we aim to detect the bounding box and category of objects in each frame. Figure 2 presents an overview of our proposed SALiency-based Input SAmpling (SALISA) framework for efficient video object detection. SALISA consists of *i)* two off-the-shelf object detection models $\mathcal{D}_{\text{key}}$ and $\mathcal{D}$, where $\text{FLOPs}_{\mathcal{D}} \ll \text{FLOPs}_{\mathcal{D}_{\text{key}}}$, *ii) a saliency map generator*, *iii) a resampling module*, and *iv) an inverse transformation module*. At inference, in the first step, we pass the first high-resolution frame $\mathbf{f}_i$ (key frame) to a high-performing detection model $\mathcal{D}_{\text{key}}$. The bounding boxes generated by this model and their corresponding scores are then passed to a saliency map generator to build a global saliency map. This

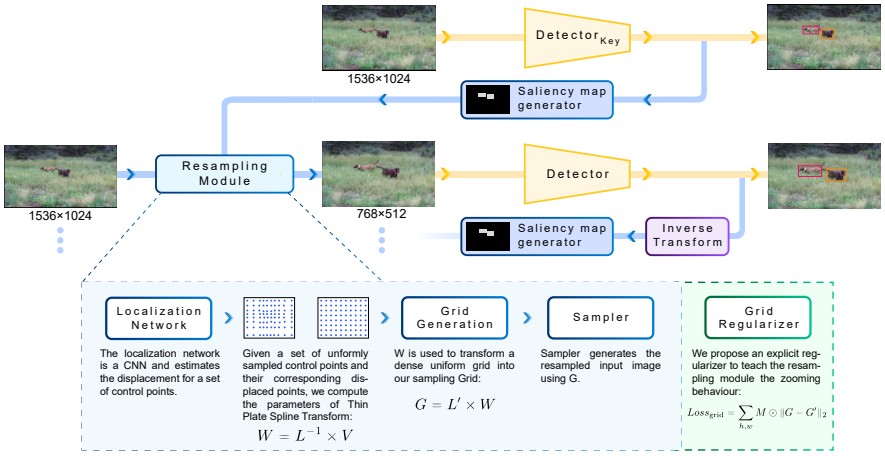

Fig. 2: **Overview of SALISA.** The first frame, from a set of high-resolution frames, is passed to a high performing detector ($\mathcal{D}_{\text{key}}$). The saliency map generator uses the prediction output to generate a saliency map. This map and the second high-resolution frame are passed to our resampling module to perform a detail-preserving down-sampling operation. The output of this module is passed to a light detector ($\mathcal{D}$) which is able to perform on par with $\mathcal{D}_{\text{key}}$ at a much lower computational cost. The output of $\mathcal{D}$ undergoes an inverse transformation to get back to the original image grid before being fed to the saliency map generation for the next frame. This process is continued for processing subsequent frames.

map and the second high-resolution frame $\mathbf{f}_{i+1}$ are then passed to our resampling module. The output of the resampling module is a down-sampled detail-preserving image $\mathbf{f}'_{i+1}$ which is fed to the light detector $\mathcal{D}$. Due to the nature of this down-sampled image, $\mathcal{D}$ is able to perform on par with $\mathcal{D}_{\text{key}}$ at a lower computational cost. For each of the following frames $\mathbf{f}_j$, we generate the saliency map from the detection output of frame $\mathbf{f}_{j-1}$ using $\mathcal{D}$. To avoid propagating errors over time, we update the detection output using the strong detector $\mathcal{D}_{\text{key}}$ at every S frames. In the following sections, we describe the different components of SALISA in details.

3.1 Saliency map generator

The saliency map generator is a non-parametric detection-to-mask generator, outputting a map corresponding to salient pixels that need to be preserved during resampling. We generate this mask from all the bounding box detections with a score above τ. The objects with an area $\alpha < 0.5\%$ of the image area are assigned a label of 1 and the ones with a larger area are assigned a label of 0.5 (we performed an ablation study on the area parameter α in Section 4.3). This will allow our resampling module to focus more on preserving the resolution of smaller objects. The background pixels are labeled as 0. Note that the saliency values of 0.5 and 1 are chosen to make a distinction between large and small objects and the exact choice of saliency values are not critical for performance. We down-sample this saliency map to 128×128 before passing it to the resampling module.

3.2 Resampling module

Our resampling module is based on a thin plate spline spatial transformer [10]. TPS-STN has three main components: *i)* The localization network, *ii)* The grid generator, and *iii)* the sampler.

Localization network. Our localization network is a VGG-style [31] architecture consisting of 10 convolutional and 2 fully connected layers (0.06 GFLOPs and 739k parameters). This network gets the saliency map as input and estimates the displacement of a set of $N = 256$ control points defined on a 16×16 grid in a Euclidean plane.

Grid generator. The grid generator is responsible for producing the sampling grid and works as follows. Given a set of N control points sampled uniformly on a 2D grid $\dot{P} \in \mathbb{R}^{N\times 2}$ and their corresponding displaced control points $\dot{V} \in \mathbb{R}^{N\times 2}$ provided by the localization network, we solve a linear system to derive the parameter $W \in \mathbb{R}^{(N+3)\times 2}$ of TPS as follows:

$$W = \underbrace{\begin{bmatrix} K & P \\ P^T & O \end{bmatrix}^{-1}}_{L} \times V, \qquad P = [\mathbf{1}, \dot{P}], \qquad V = \begin{bmatrix} \dot{V} \\ \mathbf{0} \end{bmatrix} \tag{1}$$

where the submatrix $K \in \mathbb{R}^{N\times N}$ is defined as $K_{ij} = U(\|\mathbf{p}_i, \mathbf{p}_j\|)$ where $\mathbf{p} \in \dot{P}$ and $U(r) = r^2 log(r)$ is the radial basis kernel. $O \in \mathbb{R}^{3\times 3}$, and $\mathbf{0} \in \mathbb{R}^{3\times 2}$ are submatrices of zeros and $\mathbf{1} \in \mathbb{R}^{N\times 1}$ is submatrix of ones. Note that one can precompute $L \in \mathbb{R}^{(N+3)\times(N+3)}$ and its inverse. We refer the reader to the **Appendix** for the detailed overview of the algebraic crux of the thin plate method.

Once we estimate W, we can conveniently apply the deformation to a dense uniform grid to obtain the sampling grid G, as follows:

$$G = L' \times W, \tag{2}$$

where L' is computed similarly to L but with dense points.

Sampler. In the final step, the sampler takes the sampling grid G, along with the input image $\mathbf{f}_{i+1}$ to produce the detail-preserving resampled image $\mathbf{f}'_{i+1}$. Figure 3 shows the deformation field obtained from G and the resampling results for two example images.

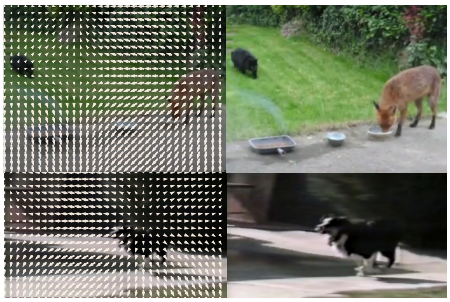

Fig. 3: **Deformation field of TPS transformation**. Left shows the deformation field overlaid on original images. Right shows the resampled images.

Regularization Learning the parameters of the localization network, without direct guidance on where to magnify, results in inhomogeneous distortions and may not preserve the desired detail. To address this, we propose to regularize the sampling grid

G through a non-parametric attention-based sampling method [38]. This resampling method takes as input a saliency map and generates a sampling grid that preserves the salient regions in the map. We propose to use the sampling grid generated by this method as a supervision signal for our sampling module to learn an explicit zooming effect. However, despite obtaining superior sampling results compared to alternative methods [26,13], this approach [38] is non-differentiable, computationally expensive, and may generate undesirable deformations when multiple objects with various saliency levels appear in the same image. This method decomposes the saliency map into two marginal distributions over x and y axes. Unfortunately, this marginalization leads to undesirable distortions for low saliency regions located on the same row/column as an object with a higher saliency level. More concretely, if the coordinates (i, j) and (i', j') in the saliency map have high values, the resulting sampling grid is not only dense at (i, j) and (i', j'), but also at (i, j') and (i', j) regardless of its saliency level. This error can be problematic when there are multiple objects with different saliency levels in the image. While our resampling module is fully differentiable and computationally inexpensive, getting an unmediated supervision from [38] may carry the same undesirable artifacts to our sampler. To address this issue, we design the following weighted ℓ_2-loss function:

$$Loss_{\text{grid}} = \sum_{h,w} M \odot \|G - G'\|_2, \tag{3}$$

where G is a grid generated by our resampling module, G' is the grid generated by the attention-based sampling method [38], and M is a weighted mask with the spatial dimension of $h \times w$. The weighted mask gets assigned different values for the small objects (O_s), large objects (O_l), and background (bg). Categorising the objects as small or large is based on the area parameter α. If the saliency map generated in Step 3.1 only contains small objects or only large objects we set (O_s, O_l, bg) to $(1, 0, \gamma)$ and $(0, 1, \gamma)$, respectively. Otherwise if it contains both small and large objects to $(1, 0, 0)$. Intuitively, when the saliency map is composed of a single saliency level (e.g., multiple small objects), [38] generates plausible zooming effects for all the objects and, therefore, we can get full supervision for the entire grid. In contrast, when the saliency map is composed of multiple saliency levels (e.g., a combination of small and large objects), the method [38] may distort objects with lower saliency. Therefore, we choose not to get supervision in those regions by masking them to zero. Note that having a down-weighted supervision (soft) in these regions did not lead to any improvements.

We train our network end-to-end by adding $Loss_{\text{grid}}$ to the detection loss. As can be seen in Figure 4, our resampling module generally generates similar zooming effects to [38] yet largely eliminates its distortions (see the flying jets and the median barrier separating the cars).

3.3 Inverse transformation module

Given the bounding box outputs of the detector D for a resampled image, we apply an inverse transformation to bring the bounding boxes coordinates back to the original image grid. This is achieved by subtracting the grid displacement offset from the bounding box coordinates. As the bounding box coordinates are floating point values,

for each bounding box coordinate, we obtain the exact original coordinate by linearly interpolating the displacements corresponding to its two closest cells on the deformation grid.

4 Experiments

To demonstrate the efficacy of SALISA, we conduct experiments on two large-scale video object detection datasets ImageNet-VID [28] and UA-DETRAC [21,20,35] as described in Section 4.1. We provide comparisons to state-of-the-art video object detection models and demonstrate that SALISA outperforms the state of the art while significantly reducing computational costs in Section 4.2. Additionally, to demonstrate the efficacy of our regularized sampling module, we compare our method with other competing sampling approaches. Finally, we present several ablation studies to discuss the effect of several design choices on the performance of our method in Section 4.3.

4.1 Experimental setup

Datasets. We evaluate our method on two large video object detection datasets: ImageNet-VID [28] and UA-DETRAC [21,20,35]. ImageNet-VID contains 30 object categories with 3862 training and 555 validation videos. Following the protocols in [17,41], during training, we also use a subset of ImageNet-DET training images, which contain the same 30 categories. We report standard mean average precision (mAP) at IoU=0.5 on the validation set, similar to [17,41]. UA-DETRAC consists of 10 hours of video (about $140k$ frames in total) captured from 100 real-world traffic scenes. The scenes include urban highways, traffic crossings, T-junctions, etc., and the bounding box annotations are provided for vehicles. The dataset comes with a partitioning of 60 and 40 videos as train and test data, respectively. Following [9], average precision (AP), averaged over multiple IoU thresholds varying from 0.5 to 0.95 with a step size of 0.05 is reported on the test data.

Implementation details. We use different variants of EfficientDet [33], namely D0-D4, as detectors in our video object detection framework. SALISA has two separate object detectors, one for the key frame ($\mathcal{D}_{\text{key}}$) and another for all succeeding frames ($\mathcal{D}$). In our experiments, we use two successive scaled-up variants of EfficientDet, for example, EfficientDet-D3 and EfficientDet-D2, where the heavier model is applied to the key frame and the lighter one to the rest of the frames. In this particular example, we refer to our model as SALISA with EfficientDet-D2. We follow the same procedure, for baseline EfficientDet models without resampling.

We first trained the resampling module independently from the detection network using the regularization loss described in 3.2. For both datasets, we then trained the EfficientDet networks, pre-trained on MS-COCO [15], in an image-based fashion. In the final step, we fine-tuned the resampling module and the object detection networks end-to-end. The complete details for training are provided in the **Appendix**.

During inference, key frames are picked once every S frames ($S = 2 \sim 32$ frames) and passed to $\mathcal{D}_{\text{key}}$ while the succeeding $S-1$ frames are processed by $\mathcal{D}$. For ImageNet-VID and UA-DETRAC experiments, we set S to 16 and 32, respectively.

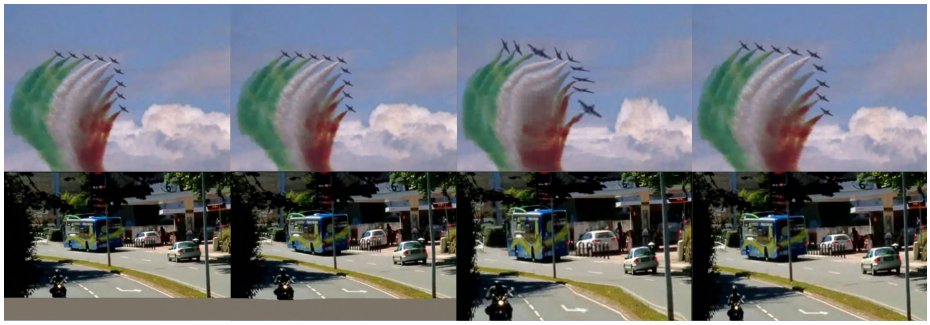

Fig. 4: **Comparison of different input resampling methods**. (a) shows example video frames from ImageNet-VID dataset. (b), (c), and (d) show the result of resampling using learning to zoom [26], TriLinear attention [38], and our proposed resampling module, respectively. Our resampling module effectively preserves the spatial resolution of salient objects and despite being regularized by [38], does not generate artifacts in background regions. This is evident from the resampled images (see the flying jets and the median barrier separating the cars).

We set the parameter of the saliency map generator τ to 0.5. We set γ controlling the weight of the regularizer in background regions to 0.5. We report the average per-frame computation cost of our model by considering the FLOPs of $\mathcal{D}_{\text{key}}$, $\mathcal{D}$, and resampling module. In our experiments, unless otherwise specified, for both baseline models and SALISA, predictions are made for odd frames f_i where $i \in \{1, 3, 5, ..., N-1\}$ and propagated to the next frame f_{i+1} without further processing. For SALISA, this means propagating the saliency maps every other frame. For both baseline and SALISA, this setup yields up to 50% reduction in FLOPs with only a small drop in the accuracy. To achieve the highest performance on each benchmark, we still apply our model densely to all frames and explicitly mention dense prediction if that is the case.

4.2 Results

Comparison to state of the art: UA-DETRAC. We compare SALISA to several image and video object detectors on the UA-DETRAC dataset: EfficientDet [33] as the state of the art in efficient object detection in images and the main baseline for SALISA, Deep Feature Flow (DFF) [41] as a seminal work on efficient object detection, and SpotNet [25] as the highest performing method on the UA-DETRAC benchmark. Figure 5 presents accuracy vs. computations trade-off curves for SALISA and the baseline EfficientDet models (D0-D3) for video object detection in UA-DETRAC. As can be seen, our method consistently outperforms the baseline EfficientDet models. Importantly, SALISA with EfficientDet-D2 (61.2%) outperforms EfficientDet-D3 model (60.3%) at lower than half the computational cost. In the low-compute regime, SALISA with EfficientDet-D0 outperforms the baseline EfficientDet-D0 model by 2.9%. The comparison with competing methods is shown in Table 1. We outperform DFF [41] both in terms of computational costs and accuracy. When densely applied to all

Table 1: Comparison with state of the art on UA-DETRAC.

Method	Backbone	mAP (%)	FLOPs (G)
DFF [41]	ResNet-50	52.6	75.3
SpotNet [25]	CenterNet [39]	62.8	972.0
EfficientDet [33]	EfficientNet-B2	59.4	5.9
EfficientDet [33]	EfficientNet-B3	60.3	13.4
SALISA(Ours)	EfficientNet-B2	61.2	5.9
SALISA(Ours)	EfficientNet-B3	62.4	13.4
EfficientDet [33]	EfficientNet-B0	51.3	1.36
EfficientDet [33]	EfficientNet-B1	56.9	3.20
SALISA(Ours)	EfficientNet-B0	54.2	1.39
SALISA(Ours)	EfficientNet-B1	59.1	3.23

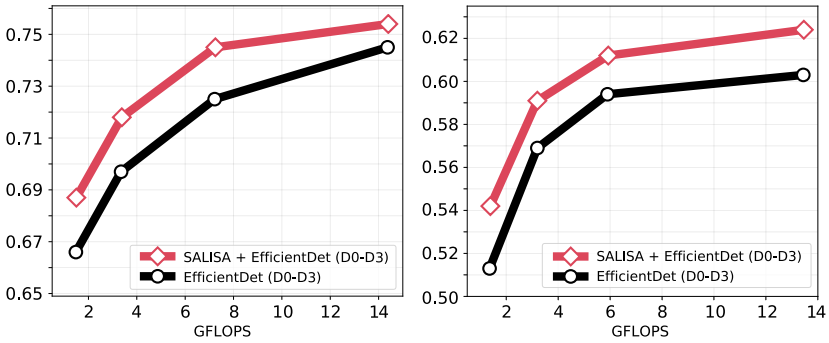

Fig. 5: Performance comparison of baseline EfficientDet [33] and corresponding SALISA+EfficientDet models on ImageNet-VID (left) and UA-DETRAC (Right).

frames, SALISA with EfficientDet-D3 achieves state-of-the-art mAP of 62.9% on UA-DETRAC at a much lower computational cost than SpotNet (972 VS. 40 GFlops).

Finally, the results presented in Figure 6 show some challenging object detection scenes from the test set. For example, the frame in the first row shows a crowded scene with many vehicles which makes zooming particularly challenging. Our model has squeezed the right side of the road to increase the resolution of the salient objects. This has enabled SALISA with EfficientDet-D1 to detect new cars which were neither detected by the baseline EfficientDet-D1 nor by the heavier keyframe detector EfficientDet-D2.

Comparison to state of the art: ImageNet-VID. The experimental results of SALISA on the ImageNet-VID dataset are presented in Table 2. We compare our method to PatchNet [23], PatchWork [4], TSM [14], DFF [41], Mobile-DFF [41], Mobile-SSD, TAFM [16], SkipConv [9], and finally EfficientDet [33] as our baseline. DFF and Mobile-DFF are flow-based methods, PatchNet is a tracking method, TAFM is an LSTM-based recurrent method, PatchWork and SkipConv conditionally limit feature computation, and TSM reduces the computation by shifting features across time. In Table 2, these methods are categorized as low compute and extremely low compute. The results show that among the extremely low compute methods, SALISA with EfficientDet-D0 significantly outperforms Mobile-SSD, PatchWork [4], TAFM [16],

Table 2: Comparison with state of the art on ImageNet-VID. * indicates that the model has been applied every three frames.

Method	Backbone	mAP (%)	FLOPs (G)
DFF (R-FCN) [41]	ResNet-101	72.5	34.9
PatchNet (R-FCN) [23]	ResNet-101	73.1	34.2
TSM [14]	ResNext101 [37]	76.3	169
SkipConv [9]	EfficientNet-B2	72.3	9.2
SkipConv [9]	EfficientNet-B3	75.2	22.4
EfficientDet [33]	EfficientNet-B2	72.5	7.2
EfficientDet [33]	EfficientNet-B3	74.5	14.4
SALISA(Ours)	EfficientNet-B2	74.5	7.2
SALISA(Ours)	EfficientNet-B3	75.4	14.4
Mobile-SSD	MobileNet-V2	54.7	2.0
PatchWork [4]	MobileNet-V2	57.4	0.97
PatchNet (EfficientDet) [23]	EfficientNet-B0	58.9	0.73
Mobile-DFF [41]	MobileNet	62.8	0.71
TAFM (SSDLite) [16]	MobileNet-V2	64.1	1.18
SkipConv [9]	EfficientNet-B0	66.2	0.98
SkipConv [9]	EfficientNet-B1	70.5	2.90
EfficientDet [33]	EfficientNet-B0	66.6	1.48
EfficientDet [33]	EfficientNet-B1	69.7	3.35
SALISA(Ours)	EfficientNet-B0*	67.4	0.86
SALISA(Ours)	EfficientNet-B0	68.7	1.50
SALISA(Ours)	EfficientNet-B1	71.8	3.38

and SkipConv [9] at a lower computational cost. While PatchNet [23] and Mobile-DFF [41] have roughly 0.15 lower GFLOPs than our lightest model, they show a significant drop in mAP ($\sim 10\%$) compared to SALISA. In general, template matching techniques offer significant computational saving without improving accuracy while SALISA can provide gains in both aspects. See **Appendix** for additional comparison with tracking baselines.

Among the low compute methods, SALISA with EfficientDet-D3 outperforms DFF and PatchNet by 2.9% and 2.3%, respectively at roughly 40% of their computational cost. TSM is the highest performing competitor that achieves an mAP of 76.3% at the cost of 169 GFLOPs. When densely applied to all frames, SALISA with EfficientDet-D3, obtains an mAP of 76.4% at 40 GFLOPs. Figure 5 presents accuracy vs. computations trade-off curves for SALISA and EfficientDet baseline models (D0-D3) for video object detection in ImageNet-VID. SALISA consistently boosts the performance of EfficientDet variants by adaptively resampling the input. Finally, SALISA with EfficientDet-D2 matches the mAP of the baseline EfficientDet-D3 at half the computational cost.

Performance across different object sizes. To demonstrate the efficacy of SALISA for detecting small objects, we report the mAP scores for different object sizes using the COCO framework [15]. As shown in Table 3, SALISA significantly improves small object detection compared to the baseline EfficientDet models. In particular, SALISA with EfficientDet-D1, improves the accuracy of small object detection by 77% (8.4% to 14.9%). Surprisingly, this is even higher than 12.5% mAP of EfficientDet-D3 baseline for small objects.

Table 3: Performance comparison (mAP) across different object sizes on UA-DETRAC.

Model	Small	Medium	Large
EfficientDet-D0	6.4	47.8	72.1
EfficientDet-D1	8.4	54.8	75.7
EfficientDet-D2	12.3	58.1	77.1
EfficientDet-D3	12.5	59.1	78.5
SALISA-D0	7.4	53.1	73.0
SALISA-D1	14.9	58.2	76.6
SALISA-D2	15.3	59.4	77.7
SALISA-D3	16.6	60.1	78.0

Table 4: Impact of input sampling method on UA-DETRAC (upper part) and ImageNet-VID (bottom part).

Resampling method	D0	D1	D2
TPS-STN [10]	51.7	57.6	60.8
Learning to zoom [26]	39.2	47.7	52.1
Trilinear attention [38]	53.6	58.7	61.5
Our resampling module	**55.6**	**61.4**	**62.7**
TPS-STN [10]	69.1	71.1	73.7
Learning to zoom [26]	69.0	71.3	74.9
Trilinear attention [38]	69.4	71.5	74.8
Our resampling module	**69.7**	**72.4**	**75.2**

The mAP of medium-sized object detection increases from 54.8 to 58.2. There is no significant change in the performance of large object sizes as the base model can also effectively detect them. That is why lighter models with smaller inputs benefit more compared to heavier models that already receive a high resolution input.

Comparison to different sampling approaches. In this experiment, we compare the performance of various sampling approaches [10,26,38] for a detail-preserving downsampling on both ImageNet-VID and UA-DETRAC datasets. To this end, we substitute our resampling module with these methods and use the same training protocol discussed in 4.1. The results are presented in Table 4. We first compare our resampling module to TPS-STN [10]. As can be seen, our regularization scheme is crucial for improving the results. TPS-STN without our regularizer, barely improves upon the baseline EfficientDet models. Our resampling module also yields a higher accuracy compared to [26] and [38] on both ImageNet-VID and UA-DETRAC datasets. While the gap in performance in different resampling methods is small on the ImageNet-VID dataset, SALISA greatly benefits from our resampling module on UA-DETRAC with a gap of more than 2% mAP. The videos in the ImageNet-VID dataset are mostly comprising one or two objects. The videos in the UA-DETRAC dataset, in contrast, include mostly crowded scenes with many objects in each frame. We conjecture that, in such multi-object wild videos the undesirable deformations induced by [26] and [38] can lower their benefits. Overall, our resampling module consistently outperforms competitors in all settings.

Wall-clock timing We report the wall-clock timing (msec) of SALISA and the baseline EfficientDet models using Nvidia Tesla-V100 32GB. The inference time of EfficientDet and SALISA for a batch size of one are as follows: **D0:** 49.4 vs 50.2, **D1:** 91.0 vs 95.4, **D2:** 152.7 vs 159.4, and **D3:** 304.8 vs 313.4. The overhead of our sampler (0.06 GFLOPs) is very small primarily because of its small input size.

4.3 Ablation study

Number of control points in TPS. Estimating the parameters of TPS relies on defining correspondences between a set of control points and their displacements. Increasing the

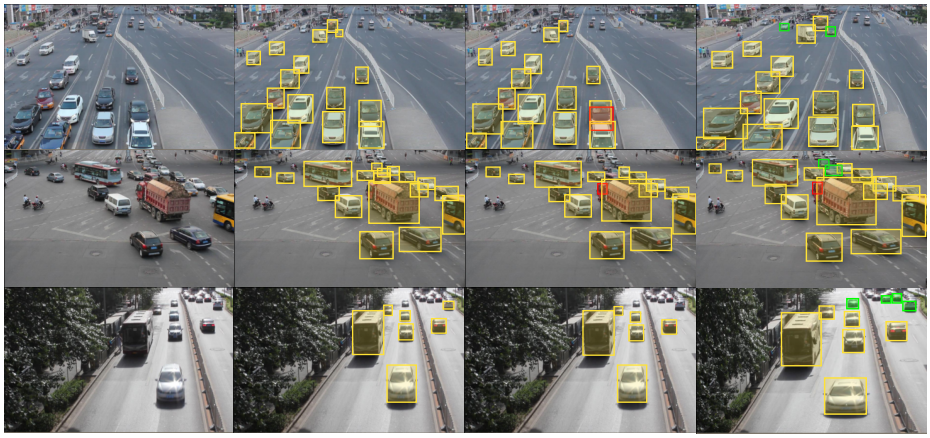

Fig. 6: **Detection results on the UA-DETRAC test set**. Yellow boxes indicate true detections, red indicates false positive detections, and green boxes refer to new detections produced by our method as a result of input resampling. (a) shows the original video frames from UA-DETRAC dataset, (b), (c), and (d) show detection results generated by EfficientDet-D2 ($\mathcal{D}_{key}$), the baseline EfficientDet-D1 detector, and SALISA with EfficientDet-D1, respectively. As can be seen from the detections in the first and third row of *(c)*, the right side of the road in the first image and the vegetation in the third image have been pushed to the side to enable magnifying salient objects. This detail-preserving down-sampling has allowed for discovery of new objects that were otherwise missed by the baseline object detector.

number of control points generally increases the flexibility of TPS. While we observe a reduction in $loss_{grid}$ when we increase the number of control points from 256 to 1024, we also notice more fluctuations and artifacts in the resulting resampled images as shown in Figure 7. By increasing the number of control points from 256 to 1024, the mAP of SALISA with EfficientDet-D0 on UA-DETRAC drops from 54.2 to 45.7, and for EfficientDet-D1 from 59.1 to 49.5. As defining 256 control points gives better detection results, we set the number of TPS control points to 256.

Robustness to keyframe detector. In this ablation, we analyze different combination possibilities for the keyframe detector ($\mathcal{D}_{key}$) and the main detector ($\mathcal{D}$) to examine the robustness of SALISA for different key detectors. The results are presented in Table 5 for different object size categories. As can be seen, there is no extra gain in medium- and large-sized object detection when combining the main detector with more expensive key frame detectors. This indicates the robustness of SALISA to the choice of keyframe detector. Note that although the mAP of small object detection improves, the additional costs of heavier networks undermine the extra gained accuracy.

Analysis of the area threshold. The resampling module gives extra focus in preserving the resolution of small object. The area threshold α determines which objects should be considered as small. Generally, we observe that a smaller value of α improves small object detection. See **Appendix** for detailed results.

	D0	D1	D2
D1	7.4	-	-
D2	7.4	14.9	-
D3	7.8	15.3	15.3

(a) Small (mAP)

	D0	D1	D2
D1	53.1	-	-
D2	53.2	58.2	-
D3	53.2	58.2	59.4

(b) Medium (mAP)

	D0	D1	D2
D1	73.0	-	-
D2	73.0	76.6	-
D3	73.0	76.6	77.7

(c) Large (mAP)

	D0	D1	D2
D1	1.39	-	-
D2	1.54	3.23	-
D3	1.98	3.67	5.96

(d) GFLOPs

Table 5: Various combinations of keyframe detector $\mathcal{D}_{key}$ (rows) and the main detector $\mathcal{D}$ (columns). (a-c) shows the mAP of various combinations for small, medium, and large object detection, respectively. (d) shows the computational costs in GFLOPs.

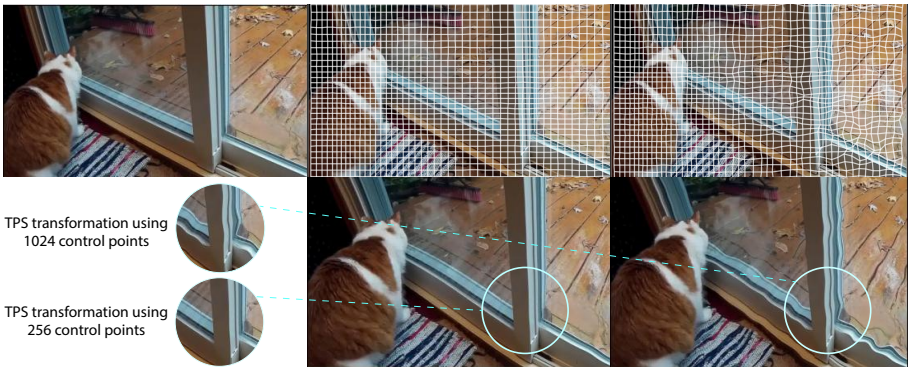

Fig. 7: **Effect of the number of control points on TPS transformation.** The top row shows the grid deformations produced by TPS with 256 (middle column) and 1024 control points (right column). The bottom row shows the corresponding resampled images.

5 Discussion and Conclusion

In this paper, we proposed SALISA, a saliency-based input sampling technique for efficient video object detection. SALISA performs a nonuniform downsampling of the input by retaining the fine-grained details of a high-resolution image while allowing for heavy downsampling of background areas. The resulting image is spatially smaller, leading to a reduction in computation costs, but preserves the important details enabling a performance comparable to a high-resolution input. We propose a novel and fully differentiable resampling module based on thin plate spline spatial transformers that generates artifact-free resampled images. SALISA achieves state-of-the-art accuracy on the ImageNet-VID and UA-DETRAC video object detection datasets in the low compute regime. In particular, it offers significant improvements in the detection of small- and medium-sized objects. A limitation of our model is that, it preserves high-resolution details by downsampling background regions more aggressively. However, when the scene is fully covered with objects, e.g. in a heavy traffic scene, proper zooming is less achievable as there is less background pixels to sub-sample.

Acknowledgements We thank Michael Hofmann, Haitam Ben Yahia, Mohsen Ghafoorian, and Ilia Karmanov for their feedback and discussions.

References

1. Avidan, S., Shamir, A.: Seam carving for content-aware image resizing. In: ACM SIGGRAPH 2007 papers, pp. 10–es (2007) 3
2. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020) 3
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020) 3
4. Chai, Y.: Patchwork: A patch-wise attention network for efficient object detection and segmentation in video streams. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3415–3424 (2019) 2, 3, 10, 11
5. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2758–2766 (2015) 2
6. Duchon, J.: Splines minimizing rotation-invariant semi-norms in sobolev spaces. In: Constructive theory of functions of several variables, pp. 85–100. Springer (1977) 2
7. Gao, J., Wang, Z., Xuan, J., Fidler, S.: Beyond fixed grid: Learning geometric image representation with a deformable grid. In: European Conference on Computer Vision. pp. 108–125. Springer (2020) 4
8. Gao, M., Yu, R., Li, A., Morariu, V.I., Davis, L.S.: Dynamic zoom-in network for fast object detection in large images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6926–6935 (2018) 3
9. Habibian, A., Abati, D., Cohen, T.S., Bejnordi, B.E.: Skip-convolutions for efficient video processing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2695–2704 (2021) 2, 3, 8, 10, 11
10. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. Advances in neural information processing systems **28**, 2017–2025 (2015) 2, 4, 6, 12
11. Jin, C., Tanno, R., Mertzanidou, T., Panagiotaki, E., Alexander, D.C.: Learning to downsample for segmentation of ultra-high resolution images. arXiv preprint arXiv:2109.11071 (2021) 4
12. Katharopoulos, A., Fleuret, F.: Processing megapixel images with deep attention-sampling models. In: International Conference on Machine Learning. pp. 3282–3291. PMLR (2019) 3
13. Li, Z., Yang, Y., Liu, X., Zhou, F., Wen, S., Xu, W.: Dynamic computational time for visual attention. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 1199–1209 (2017) 7
14. Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient video understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7083–7093 (2019) 10, 11
15. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014) 2, 8, 11
16. Liu, M., Zhu, M.: Mobile video object detection with temporally-aware feature maps. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5686–5695 (2018) 10, 11
17. Liu, M., Zhu, M., White, M., Li, Y., Kalenichenko, D.: Looking fast and slow: Memory-guided mobile video object detection. arXiv preprint arXiv:1903.10172 (2019) 2, 3, 8
18. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV (2016) 3

19. Luo, H., Xie, W., Wang, X., Zeng, W.: Detect or track: Towards cost-effective video object detection/tracking. In: AAAI (2019) 3
20. Lyu, S., Chang, M.C., Du, D., Li, W., Wei, Y., Del Coco, M., Carcagnì, P., Schumann, A., Munjal, B., Choi, D.H., et al.: Ua-detrac 2018: Report of avss2018 & iwt4s challenge on advanced traffic monitoring. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–6. IEEE (2018) 8
21. Lyu, S., Chang, M.C., Du, D., Wen, L., Qi, H., Li, Y., Wei, Y., Ke, L., Hu, T., Del Coco, M., et al.: Ua-detrac 2017: Report of avss2017 & iwt4s challenge on advanced traffic monitoring. In: Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on. pp. 1–7. IEEE (2017) 8
22. Mao, H., Kong, T., Dally, W.J.: Catdet: Cascaded tracked detector for efficient object detection from video. arXiv preprint arXiv:1810.00434 (2018) 2, 3
23. Mao, H., Zhu, S., Han, S., Dally, W.J.: Patchnet–short-range template matching for efficient video processing. arXiv preprint arXiv:2103.07371 (2021) 2, 3, 10, 11
24. Marin, D., He, Z., Vajda, P., Chatterjee, P., Tsai, S., Yang, F., Boykov, Y.: Efficient segmentation: Learning downsampling near semantic boundaries. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2131–2141 (2019) 4
25. Perreault, H., Bilodeau, G.A., Saunier, N., Héritier, M.: Spotnet: Self-attention multi-task network for object detection. CRV (2020) 9, 10
26. Recasens, A., Kellnhofer, P., Stent, S., Matusik, W., Torralba, A.: Learning to zoom: a saliency-based sampling layer for neural networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 51–66 (2018) 4, 7, 9, 12
27. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28**, 91–99 (2015) 3
28. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015) 8
29. Setlur, V., Takagi, S., Raskar, R., Gleicher, M., Gooch, B.: Automatic image retargeting. In: Proceedings of the 4th international conference on Mobile and ubiquitous multimedia. pp. 59–68 (2005) 3
30. Shen, Y., Wu, N., Phang, J., Park, J., Kim, G., Moy, L., Cho, K., Geras, K.J.: Globally-aware multiple instance classifier for breast cancer screening. In: International Workshop on Machine Learning in Medical Imaging. pp. 18–26. Springer (2019) 3
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) 6
32. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. pp. 6105–6114. PMLR (2019) 2, 3
33. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10781–10790 (2020) 2, 3, 8, 9, 10, 11
34. Thavamani, C., Li, M., Cebron, N., Ramanan, D.: Fovea: Foveated image magnification for autonomous navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15539–15548 (2021) 4
35. Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M., Qi, H., Lim, J., Yang, M., Lyu, S.: UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. Computer Vision and Image Understanding (2020) 8
36. Xia, F., Wang, P., Chen, L.C., Yuille, A.L.: Zoom better to see clearer: Human and object parsing with hierarchical auto-zoom net. In: European Conference on Computer Vision. pp. 648–663. Springer (2016) 3

37. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1492–1500 (2017) 11
38. Zheng, H., Fu, J., Zha, Z.J., Luo, J.: Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5012–5021 (2019) 3, 4, 7, 9, 12
39. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019) 10
40. Zhu, X., Dai, J., Yuan, L., Wei, Y.: Towards high performance video object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7210–7218 (2018) 2, 3
41. Zhu, X., Xiong, Y., Dai, J., Yuan, L., Wei, Y.: Deep feature flow for video recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2349–2358 (2017) 2, 3, 8, 9, 10, 11
42. Zoph, B., Cubuk, E.D., Ghiasi, G., Lin, T.Y., Shlens, J., Le, Q.V.: Learning data augmentation strategies for object detection. In: European Conference on Computer Vision. pp. 566–583. Springer (2020) 2, 3