ANT: Adapt Network Across Time for Efficient Video Processing

Feng Liang[1] Ting-Wu Chin[2] Yang Zhou[1] Diana Marculescu[1]
[1] The University of Texas at Austin [2] Carnegie Mellon University
{jeffliang, yangzhou25672, dianam}@utexas.edu tingwuc@alumni.cmu.edu

Abstract

Abundant redundancies exist in video streams, thereby pointing to opportunities to save computations. Towards this end, we propose the Adaptive Network across Time (ANT) framework to harness these redundancies for reducing the computational cost of video processing. Unlike most dynamic networks that adapt their structures to different static inputs, our method adapts networks along the temporal dimension. By inspecting the semantic differences between frames, the proposed ANT chooses a purpose-fit network at test time to reduce overall computation, i.e.*, switching to a smaller network when observing mild differences. The proposed ANT adapts the structured networks within a supernet, making it hardware-friendly and therefore achieves actual acceleration in real-world scenarios. The proposed ANT is powered by (1) a fusion module that utilizes the past features and (2) a dynamic gate to adjust the network in a predictive fashion with negligible extra cost. To ensure the generality of each subnet and the gate's fairness, we propose a two-stage training scheme. We first train a weight-sharing supernet and then jointly train fusion modules and gates. Evaluation of the video detection task with the modern EfficientDet reveals the effectiveness of our approach.*

1. Introduction

Deep learning has come to a mobile era where we need to deploy machine learning models on common mobile platforms such as smartphones, drones, and self-driving vehicles. A series of efficient deep learning algorithms have been proposed to achieve this goal, such as network pruning [5, 6, 19, 23, 24], quantization [3, 7, 9, 11, 15], Neural Architecture Search (NAS) [2, 8, 10, 22, 28, 34, 40], and dynamic inference [4, 14, 18, 31, 35]. Dynamic inference methods have attracted much attention because of their ability to save computation via adapting networks according to different inputs, *i.e.*, using fewer computations for 'easy' samples. However, most dynamic networks are limited to static inputs. This paper studies the notion of dynamic network for streaming applications, such as video processing.

Video processing usually involves a significantly larger volume of data compared to static images. The computation cost grows linearly with the number of input frames for deep neural networks. Thus, it is not practical for mobile devices to process each frame, especially for dense prediction tasks, such as object detection. Existing approaches [12, 16, 20, 27, 39] propose to exploit the temporal redundancies across frames via feature propagation [39] or sparse convolution [12]. However, such approaches may not achieve actual acceleration in real-world scenarios because of additional optical flow extraction [16, 39] or it requires dedicated sparse convolution implementation [12].

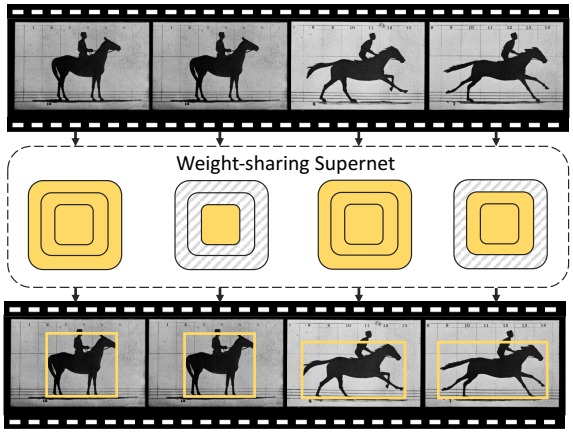

Figure 1. ANT illustrated with *The Horse in Motion* [25]. The proposed ANT adapts a purpose-fit network by inspecting the semantic differences between frames, *i.e.*, switching to a smaller network when observing mild differences so as to save computation.

To address the aforementioned issues, we propose a framework which builds a dynamic network, specifically, a weight-sharing supernet, that could **A**dapt **N**etworks across **T**ime. As denoted in Fig. 1, ANT switches to a smaller network when observing mild differences between frames to save overall computation. The differences are measured in a semantic space because pixels may change rapidly from frame to frame, but the semantic content of a scene evolves more slowly. Thanks to the structured property of the supernet [2, 18, 36, 37], ANT is hardware-friendly and can achieve actual speedup in common devices without the needs for dedicated convolution implementations.

The success of the proposed ANT lies in two key factors: (1) leveraging the information of past frames, and (2) adapting networks with negligible extra cost. Unlike prior art [12, 39] which uses potentially expensive feature propagation, ANT adopts a hardware-friendly feature fusion to leverage past frames, *e.g.*, a concatenation followed by a convolution. In order to adapt networks across time, not only does ANT rely on the current frame, but it also relies on the differences across frames. Specifically, we propose a dynamic gate [18, 31] operated on features after fusion to predict the network. The $\texttt{gumbel} - \texttt{softmax}$ function [17] is utilized to optimize the non-differentiable dynamic gates.

In ANT, we need to train supernet weights as well as fusion modules and dynamic gates, which is a highly entangled bi-level optimization problem. To ensure the generality of each subnet and the gate's fairness, we propose using a two-stage training scheme. We first train a weight-sharing supernet following the weight-sharing NAS [2, 36]. In the second stage, we jointly train fusion modules and dynamic gates.

We evaluate ANT on a challenging video object detection task [33] with the state-of-the-art mobile object detector EfficientDet [30]. ANT is able to achieve nearly 30% speedup compared to a static object detector, with negligible accuracy drop. Experiments also show that our ANT is superior to its non-temporal dynamic network counterpart as far as efficiency is concerned.

2. Related Work

Dynamic networks. Dynamic networks can adapt their structures to different inputs, leading to notable computational efficiency [13]. Most literature [18, 31, 32, 35] is in the static image space, where 'easy' samples are routed to fewer computations to amortize the cost. Strategies include early exit [14], dropping residual blocks [35], or selecting a fraction of network [18, 31]. In contrast, the proposed ANT is a dynamic network across the temporal dimension. Our ANT framework adapts networks to the current frame *and* the differences across frames, leveraging abundant redundancies in video streams to save computations.

Efficient video processing. Video can be viewed as a consecutive of frames. The key of efficient video processing is in exploiting temporal redundancies across frames. A common strategy is feature propagation [12, 20, 38, 39], which computes the expensive backbone features only on key frames. Subsequent non-key frames then adapt the backbone features from key-frames directly [27] or after spatial alignments via optical flow [16, 39], dynamic filters [20], or sparse convolution [12]. Similarly, ANT also propagates features from the past key frames. However, we use a hardware-friendly fusion module without expensive optical flow [16, 39] or dedicated sparse convolution [12].

3. Methodology

The proposed ANT is a weighting-sharing supernet with dynamic gates to adapt purpose-fit networks according to differences between frames. In this section, we first discuss the weight-sharing supernet training (Sec. 3.1), then the joint training of our ANT (Sec. 3.2), finally the adaptive key frame selection(Sec. 3.3).

3.1. Weight-sharing supernet training

We apply ANT to the mobile object detector EfficientDet [30], which consists of a backbone feature extractor, a Bi-FPN neck, and a detection head. We only transfer the backbone into a supernet because it accounts for most of the computation.

Let the weights of the backbone supernet be W_o and a subnet configuration be $\{arch\}$, we denote the sliced weights for the subnet by $W_o(arch)$ The overall training objective is to optimize W_o to make each supported subnet $W_o(arch)$ achieve the same level of accuracy as the independently trained network with the same architectural configuration. To make our ANT hardware-friendly, we only select different filter numbers (widths) in the supernet. We first train a standalone network on the target dataset and use it as the initialization of supernet as in [2]. Knowledge distillation [2] and sandwich rule [36,37] are also used to boost the accuracy.

3.2. Joint training fusions and gates

The overview of ANT is depicted in Fig. 2. Key frames are processed through the entire network. Unlike SkipConv [12], which stores features of every layer, we only store the stage-level features [30], *i.e.* the last-layer feature with the same spatial resolution. Non-key frames would go through a subnet and reuse the stage-level features of key frames via a light fusion module. More specifically, we first concatenate corresponding features and then perform a 1×1 point-wise convolution followed by a 3×3 depth-wise convolution. There are normalization and activation layers following each convolution. We have also tried other fusion mechanisms, such as self-attention [1]. It turns out self-attention [1] can increase the performance but brings considerable latency workload. Thus we use the simple convolutional fusion as our default setting.

Fused stage-level features, containing the current frames and frames differences will decide the stage width through a dynamic gate. For the dynamic gate, we follow the standard design from prior work [18, 31]. To reduce complexity, every stage-level feature is first condensed to a vector via a global pooling. Then, we use the pooled vector to predict the width of the current sample. Following [18], we adopt two fully connected layers and a ReLU non-linearity layer in between to predict scores for each stage width. An

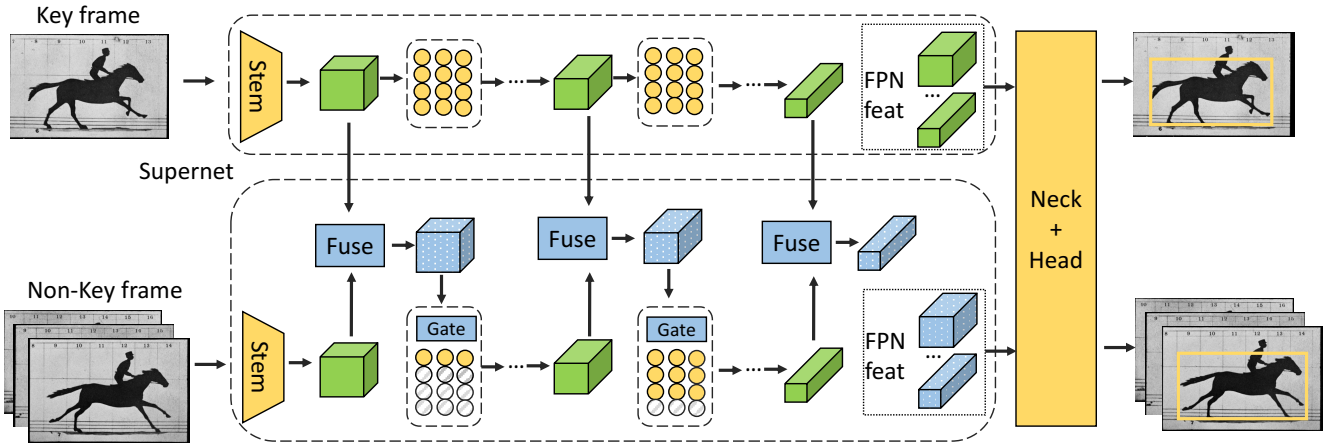

Figure 2. Overview of our ANT. Key frames would go through the entire network. Non-key frames would reuse the stage-level features of key frames via a light fusion module. For non-key frames, fused stage-level features, containing the information of current frame *and* the changes across frames would decide the stage width through a dynamic gate. The neck and head are shared for all frames.

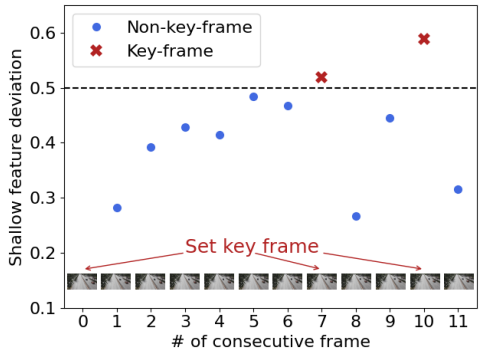

Figure 3. Adaptive key frame selection. As we proceed further away from the key frame, the feature deviation gradually increases. When the deviation goes beyond a pre-defined threshold, the current frame will be selected as a new key frame.

`argmax` function is subsequently applied to generate the predicted choice. The $\texttt{gumbel} - \texttt{softmax}$ [17] is utilized to optimize the non-differentiable dynamic gates.

After we obtain the static image supernet from Sec. 3.1, we add the random initialized fusion module, dynamic gates to the framework, and jointly train the weights. We freeze the stem module as shown in Fig. 2 and use a fixed key/non-key frame scheduler, *e.g.*, 1/3 in our experiment. We still use the knowledge distillation (KD) technique in this stage. Specifically, the non-key frames will also go through the entire work to get the golden features(not shown in Fig. 2). The KD loss is applied between golden features and fused features. Intuitively, KD helps guide the fusion module learning.

3.3. Adaptive key frame selection

An important video processing step is to select the key frames, *i.e.*, the frame that goes through the entire network and serves as a reference. More frequent key frames would lead to better performance but result in more computation cost. To make our ANT more flexible, we abandon the fixed-rate schedulers [12, 39] and choose to select the key frame adaptively during test. While pixels may change rapidly from frame to frame, we find that the semantic content of a scene evolves more slowly. Moreover, the deviation of shallow features, *e.g.*, features after first conv-stem module, are good to represent the semantic differences across frames [20]. We calculate the normalized L_1 deviation across different frames in Fig. 3. We can easily find that the feature deviation would generally increase over time. When the deviation reaches to a certain threshold, we reset the current frame as the new key frame and continue the process.

4. Experiments

4.1. Setup

Dataset. We evaluate the proposed ANT on a challenging video object detection benchmark UA-DETRAC dataset [33]. The dataset consists of 10 hours of real-world traffic videos at 24 different locations, leading to 140 thousand frames and a total of 1.21 million labeled bounding boxes of vehicles. The total 100 videos are split 60/40 as rain and test data, respectively. The performance is evaluated in terms of average precision (AP) across multiple IoU thresholds $[0.5 : 0.05 : 0.95]$, similar to COCO [21].

Model. We use the state-of-the-art mobile object detector EfficientDet [30]. As detailed in Sec. 3.1, we only turn the EfficientNet [29] backbone into supernet stages. The basic building block of EfficientNet is the mobile inverted bottleneck MBConv [26]. For simplicity, we denote by MBConv$K$ the MBConv with expansion ratio K. We modify 6 MBConv6 stages into supernet, excluding the first stem convolution and second MBConv1 block. For every

Method	Params (M)	GMAC	Latency† (ms)	AP
D0 [12, 29]	**3.8**	2.2	320	52.7
D0 w/ SkipConv [12]	4.0	**0.7**	480	52.3
D0 w/ ANT (Ours)	4.3	1.7	262	52.5
D0+ w/ ANT (Ours)	5.5	1.6	**251**	**52.8**

Table 1. Comparison with state-of-the-art methods of object video detection on UA-DETRAC. D0 is the abbreviation of EfficientDet-D0. The proposed ANT is hardware-friendly and can achieve real acceleration on common devices. D0+ denotes using wider backbone. † Latency is measured on one intel i7 CPU core.

stage, we have 6 expansion ratios $[1, 2, 3, 4, 5, 6]$, leading to 6^6 subnets.

Implementation details We follow the UA-DETRAC setting of Skip-Conv [12]. The standalone network is initialized with the pre-trained weights from MS COCO [21]. We train the model for 4 epochs using the SGD optimizer with momentum 0.9 and weight decay 4e-5. The initial learning rate is 0.01 and decays to 0.001 at epoch 3. The batch size is 16 across 2 GPUs (batch size 8 for each card). We also use a high drop path rate of 0.5 to avoid overfitting. For the first stage of supernet training, we initialize the supernet with the standalone network and follow the same settings as the standalone network. We initialize with the trained supernet for the second stage of joint training and add random initialized fusion modules and gates. Then we train ANT for 8 epochs with a lower learning rate of 0.002.

4.2. Main results

Comparison with state-of-the-art methods. In order to run the detector on edge devices, we use the smallest EfficientDet-D0 for our experiments. The state-of-the-art method SkipConv [12] achieves a considerable reduction of Giga-multiply-accumulate (GMAC). However, when we measure its latency on CPU, which has no sparse convolution implementation, we find SkipConv is *slower* than the regular network (see Tab. 1). In contrast, the proposed ANT is hardware-friendly and can achieve actual acceleration in real-world scenarios. More specifically, ANT can reduce the amortized latency of EfficientDet-D0 from 320ms to 262ms with negligible accuracy loss. Inspired by NAS literature [2, 36], we further use a wider backbone to boost the accuracy. We widen all the channels in D0's backbone with a 1.1× factor and apply ANT on it (denoted as D0+). With a wider backbone, ANT reduces the latency from 320ms to 251ms (1.3× reduction) with a slight AP increase from 52.7 to 52.8. Our results show that GMAC counts are poor proxies for the latency of ML tasks on real hardware.

Comparison with static NAS models. As detailed in Sec. 3.1, we first start with a weight-sharing supernet. We can easily sample static models from the supernet using a standard NAS procedure [36]. As shown in Fig. 4, the proposed dynamic ANT performs consistently better than static NAS models. This is because the proposed ANT can adapt the network across time, while static NAS models are fixed for all the inputs. Moreover, we may have to maintain several different static networks for different computation constraints. The proposed ANT, in contrast, can easily fit into different requirements through adjusting the key/non-key frame threshold (Sec. 3.3).

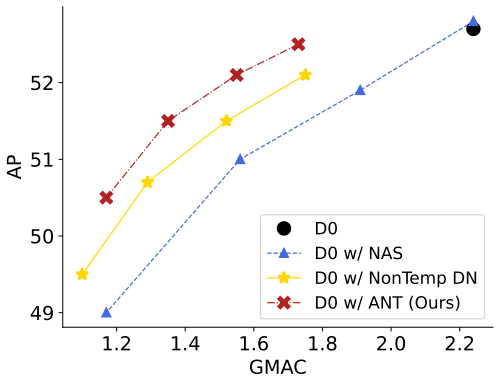

Figure 4. Ablation study. D0 is the abbreviation of EfficientDet-D0. D0 w/ NAS are static networks that are searched using NAS. D0 w/ NonTemp DN is the non-temporal dynamic network that changes the network solely depending on the current frame. The proposed ANT adapts networks depending on the current frame and the differences across frames.

Comparison with the non-temporal dynamic network. We further compare with the non-temporal dynamic network [18] that changes network solely depending on the current frame. As shown in Fig. 4, the non-temporal dynamic network suppresses static NAS models but performs worse than the proposed ANT. The main reason is that not only can ANT use the current frame, but it can also utilize the differences across frames. The proposed ANT can leverage abundant redundancies in video streams.

5. Conclusion

Towards the goal of efficient video processing, we propose the Adaptive Network across Time (ANT) framework to harness redundancies in video streams so as to save computations. The proposed ANT adapts a purpose-fit network by inspecting the semantic differences between frames, *i.e.*, switching to a smaller network when observing mild differences. ANT is built upon a weight-sharing supernet with proposed fusion and dynamic gate modules. ANT is hardware-friendly and can achieve actual acceleration in real-world scenarios.

Acknowledgement

This research was supported in part by NSF CCF Grant No. 2107085 and NSF CSR Grant No. 1815780.

References

[1] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3286–3295, 2019. 2

[2] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019. 1, 2, 4

[3] Ting-Wu Chin, Pierce I-Jen Chuang, Vikas Chandra, and Diana Marculescu. One weight bitwidth to rule them all. In *European Conference on Computer Vision*, pages 85–103. Springer, 2020. 1

[4] Ting-Wu Chin, Ruizhou Ding, and Diana Marculescu. Adascale: Towards real-time video object detection using adaptive scaling. *Proceedings of Machine Learning and Systems*, 1:431–441, 2019. 1

[5] Ting-Wu Chin, Ruizhou Ding, Cha Zhang, and Diana Marculescu. Towards efficient model compression via learned global ranking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1518–1528, 2020. 1

[6] Ting-Wu Chin, Cha Zhang, and Diana Marculescu. Layer-compensated pruning for resource-constrained convolutional neural networks. *arXiv preprint arXiv:1810.00518*, 2018. 1

[7] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016. 1

[8] Xiaoliang Dai, Peizhao Zhang, Bichen Wu, Hongxu Yin, Fei Sun, Yanghan Wang, Marat Dukhan, Yunqing Hu, Yiming Wu, Yangqing Jia, et al. Chamnet: Towards efficient network design through platform-aware model adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11398–11407, 2019. 1

[9] Ruizhou Ding, Ting-Wu Chin, Zeye Liu, and Diana Marculescu. Regularizing activation distribution for training binarized deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11408–11417, 2019. 1

[10] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017, 2019. 1

[11] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. *arXiv preprint arXiv:2103.13630*, 2021. 1

[12] Amirhossein Habibian, Davide Abati, Taco S Cohen, and Babak Ehteshami Bejnordi. Skip-convolutions for efficient video processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2695–2704, 2021. 1, 2, 3, 4

[13] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[14] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens Van Der Maaten, and Kilian Q Weinberger. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*, 2017. 1, 2

[15] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018. 1

[16] Samvit Jain, Xin Wang, and Joseph E Gonzalez. Accel: A corrective fusion network for efficient semantic segmentation on video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8866–8875, 2019. 1, 2

[17] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 2, 3

[18] Changlin Li, Guangrun Wang, Bing Wang, Xiaodan Liang, Zhihui Li, and Xiaojun Chang. Dynamic slimmable network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8607–8617, 2021. 1, 2, 4

[19] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016. 1

[20] Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5997–6005, 2018. 1, 2, 3

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 3, 4

[22] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018. 1

[23] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pages 2736–2744, 2017. 1

[24] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018. 1

[25] E Muybridge. The horse in motion. library of congress prints and photographs division, 2017. 1

[26] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 3

[27] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. Clockwork convnets for video semantic segmentation. In *European Conference on Computer Vision*, pages 852–868. Springer, 2016. 1, 2

[28] Dimitrios Stamoulis, Ruizhou Ding, Di Wang, Dimitrios Lymberopoulos, Bodhi Priyantha, Jie Liu, and Diana Marculescu. Single-path nas: Designing hardware-efficient convnets in less than 4 hours. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 481–497. Springer, 2019. 1

[29] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 3, 4

[30] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 2, 3

[31] Andreas Veit and Serge Belongie. Convolutional networks with adaptive inference graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–18, 2018. 1, 2

[32] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 409–424, 2018. 2

[33] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu. Ua-detrac: A new benchmark and protocol for multi-object detection and tracking. *Computer Vision and Image Understanding*, 193:102907, 2020. 2, 3

[34] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10734–10742, 2019. 1

[35] Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S Davis, Kristen Grauman, and Rogerio Feris. Blockdrop: Dynamic inference paths in residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8817–8826, 2018. 1, 2

[36] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Thomas Huang, Xiaodan Song, Ruoming Pang, and Quoc Le. Bignas: Scaling up neural architecture search with big single-stage models. In *European Conference on Computer Vision*, pages 702–717. Springer, 2020. 1, 2, 4

[37] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. *arXiv preprint arXiv:1812.08928*, 2018. 1, 2

[38] Xizhou Zhu, Jifeng Dai, Lu Yuan, and Yichen Wei. Towards high performance video object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7210–7218, 2018. 2

[39] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2349–2358, 2017. 1, 2, 3

[40] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018. 1