LightTrack: Finding Lightweight Neural Networks for Object Tracking via One-Shot Architecture Search —— Supplementary Material ——

Bin Yan[1,2,*], Houwen Peng[1,*,†], Kan Wu[1,3,*], Dong Wang[2], Jianlong Fu[1], and Huchuan Lu[2,4]
[1]Microsoft Research Asia [2]Dalian University of Technology
[3]Sun Yat-sen University [4]Peng Cheng Laboratory

In this supplementary material, we present more technical details of the proposed method (Section 4 in the manuscript) and more implementation details and experimental comparisons (Section 5). We also provide the changing of architectures during search. The main content is summarized as follows.

- **(Section 4) LightTrack**
  We elaborate the "Phase 2: Training Tracking Supernet" and "Phase 3: Searching with Evolutionary Algorithm" of the search pipeline in Appendix A and Appendix B, respectively.
- **(Section 5) Experiments**
  In Appendix C, we provide the details of the search space and the supernet architecture used to search for LightTrack-LargeA and LargeB.
- **(Others) Searched Architectures**
  We visualize the changes of architectures during search in the Archs.gif.

Appendix A

We present the the details of tracking supernet training in Alg. 1.

Algorithm 1 Training Tracking Supernet

Input: Tracking training set T_{trk} consisting of exemplar images and search images, backbone supernet $\mathcal{N}_b$ with search space $\mathcal{A}_b$ and pretrained weight W_b^p, head supernet $\mathcal{N}_h$ with search space $\mathcal{A}_h$ and weight W_h, max iteration I.
Output: Trained tracking supernet $\mathcal{N}(\mathcal{A}_b, W_b^*; \mathcal{A}_h, W_h^*)$.

Initialize the weight W_b of the backbone supernet with the pre-trained weight W_b^p.
Initialize the head supernet with random weight W_h.
$i \leftarrow 0$.
while train iteration $i \in [0, I)$ **do**
    Randomly sample an architecture $\alpha = \{\alpha_b, \alpha_h\}$ from the tracking supernet $\mathcal{N} = \{\mathcal{N}_b, \mathcal{N}_h\}$.
    Run the subnet $\mathcal{N}(\alpha_b, W_b; \alpha_h, W_h)$ over the current training batch, calculate the loss $\mathcal{L}_{train}^{trk}$.
    Update the subnet's weight $\{W_b(\alpha_b), W_h(\alpha_h)\}$ using the gradients with respect to $\mathcal{L}_{train}^{trk}$.
    $i \leftarrow i + 1$
end while
Output the trained tracking supernet $\mathcal{N}$ with the optimal weight $\{W_b^*, W_h^*\}$.

Appendix B

Alg. 2 presents the evolutionary algorithm used in LightTrack.

Appendix C

Tab. 1 shows the search space and the supernet architecture used to search for LightTrack-LargeA and LightTrack-LargeB. The basic building blocks are still the inverted residual block (MBConv) [4] with squeeze-excitation module [3, 2] and the depthwise separable convolution (DSConv) [1]. However, different from the settings of LightTrack-Mobile, this large

Algorithm 2 Searching with Evolutionary Algorithm

Input: Trained tracking supernet $\mathcal{N}$, tracking training subnet T_{trk}^{sub}, tracking validation set V_{trk}, maximum population size $|P|_{max}$, number of dominant individuals K, mutation probability $prob_{mut}$, user-specified maximum Flops $Flops_{max}$ and Params $Params_{max}$, number of evolution generation G_E.

Output: The most promising architecture α^*.

$pop_{(0)} \leftarrow$ `get-population`$(|P|_{max},\ Flops_{max}, Params_{max})$, $\alpha^* \leftarrow \emptyset$, $g \leftarrow 0$
while $g \in [0, G_E)$ **do**
  while $\alpha \in pop_{(g)}$ **do**
    Recalculate subnet $\mathcal{N}(\alpha, W)$'s BN on T_{trk}^{sub}.
    Compute tracking accuracy on V_{trk}.
  end while
  Get dominant individuals $pop_{(g)}^{dom}$ by selecting top K individuals with the highest tracking accuracy.
  Get mutation and crossover population $pop_{(g)}^{mut}$, $pop_{(g)}^{crs}$.

$$\begin{aligned} pop_{(g)}^{mut} &\leftarrow \texttt{mutation}(\ pop_{(g)}^{dom},\ prob_{mut},\ constraints\ ), \\ pop_{(g)}^{crs} &\leftarrow \texttt{crossover}(\ pop_{(g)}^{dom},\ constraints\ ), \end{aligned} \tag{1}$$

  $pop_{(g+1)} \leftarrow \{pop_{(g)}^{mut}, pop_{(g)}^{crs}\}$.
  $g \leftarrow g + 1$.
end while
Output the optimal architecture α^* by choosing the subnet with the highest tracking accuracy.

supernet includes more searchable layers in backbone (21 *v.s.* 14 layers) and in head (16 *v.s.* 8 layers). Besides, there are more channels ($\{192,256,320\}$ *v.s.* $\{128,192,256\}$) in the head supernet.

	Input Shape	Operators	$N_{choices}$	Chn	Rpt	Stride
Backbone	$256^2 \times 3$	3×3 Conv	1	16	1	2
	$128^2 \times 16$	DSConv	1	16	1	1
	$128^2 \times 16$	MBConv / Skip	6	24	**5**	2
	$64^2 \times 24$	MBConv / Skip	6	40	**5**	2
	$32^2 \times 40$	MBConv / Skip	6	80	**5**	2
	$16^2 \times 80$	MBConv / Skip	6	96	**6**	1
Cls Head	$16^2 \times 128$	DSConv	6	C_1	1	1
	$16^2 \times C_1$	DSConv / Skip	3	C_1	**15**	1
	$16^2 \times C_1$	3x3 Conv	1	1	1	1
Reg Head	$16^2 \times 128$	DSConv	6	C_2	1	1
	$16^2 \times C_2$	DSConv / Skip	3	C_2	**15**	1
	$16^2 \times C_2$	3x3 Conv	1	4	1	1

Table 1: Search space and supernet structure of LightTrack-LargeA and LightTrack-LargeB. "$N_{choices}$" represents the number of choices for the current block. "Chn" and "Rpt" denote the number of channels per block and the maximum number of repeated blocks in a group, respectively. "Stride" indicates the convolutional stride of the first block in each repeated group. The classification and regression heads are allowed to use different numbers of channels, denoted as $C_1, C_2 \in \{192, 256, 320\}$. The input is a search image with size of 256×256×3.

References

[1] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 1

[2] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019. 1

[3] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 1

[4] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 1