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

Bin Yan*, Houwen Peng*, Kan Wu*, Dong Wang, Jianlong Fu, Huchuan Lu

Paper link: https://arxiv.org/abs/2104.14545v1
Code link: https://github.com/researchmm/LightTrack

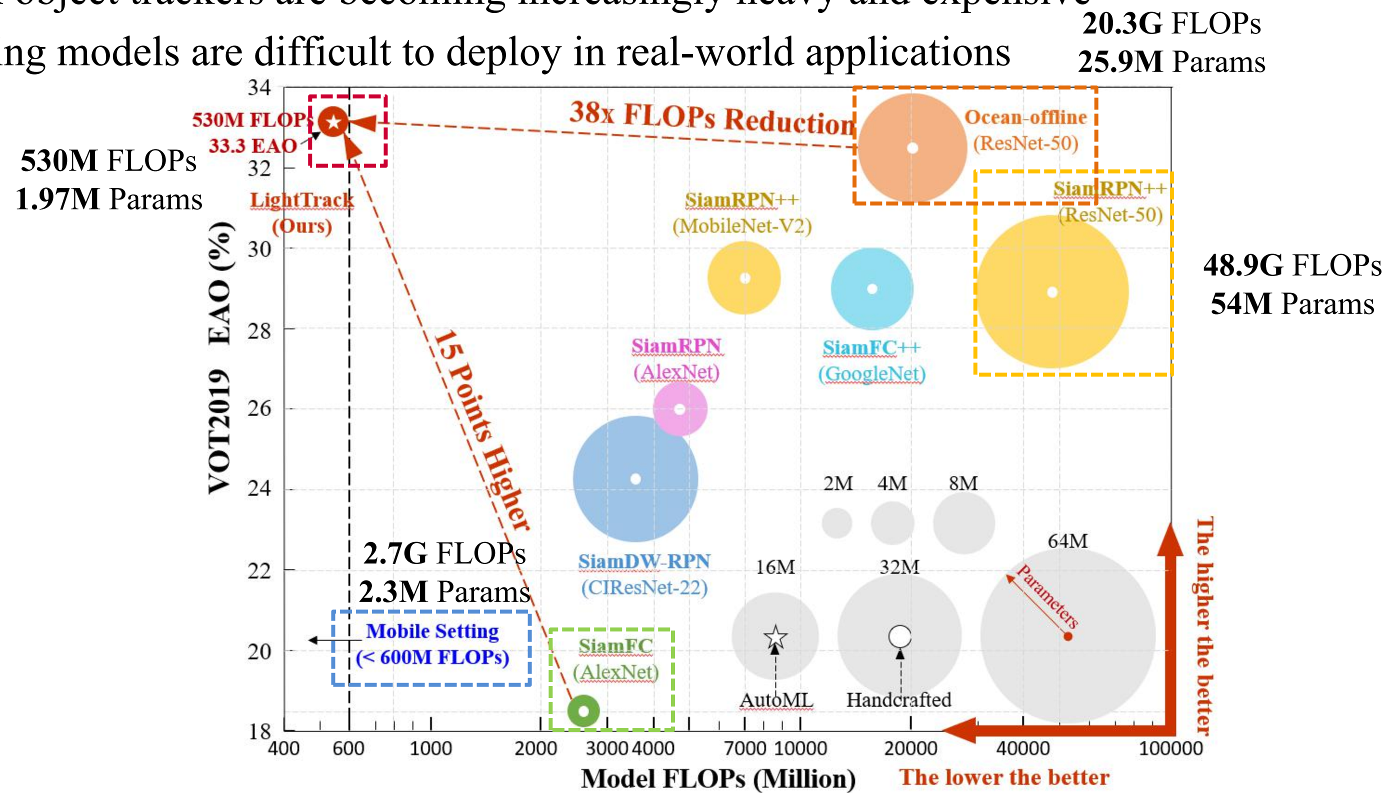
Overview

- Motivation
- Potential Solutions for Lightweight model design
- Introduction to Neural Architecture Search
- LightTrack
- Experiments

LightTrack

Motivation

- SOTA object trackers are becoming increasingly heavy and expensive
- Tracking models are difficult to deploy in real-world applications



LightTrack

Potential Solutions

- Model Compression
  - bringing non-negligible performance degradation ×
- Handcraft new compact and efficient models
  - engineering expensive & heavily relying on human expertise and experience ×
- Automating the design of lightweight models with NAS
  - Automatically designing optimal architectures on resource-limited hardware platforms for object tracking task √

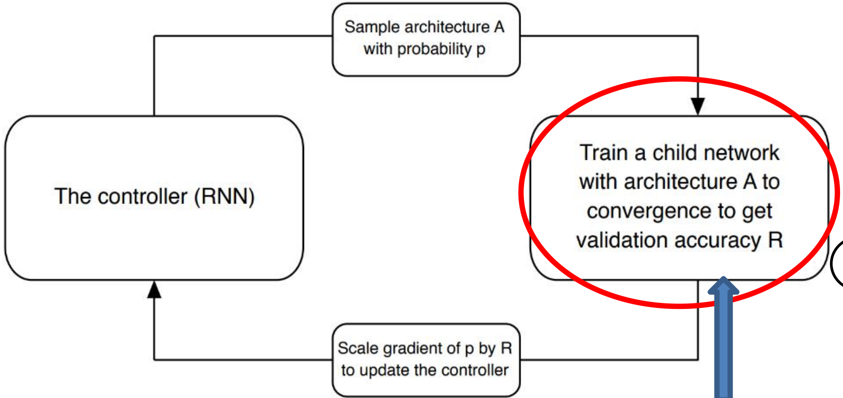
History and early NAS methods

The simplest idea for NAS:

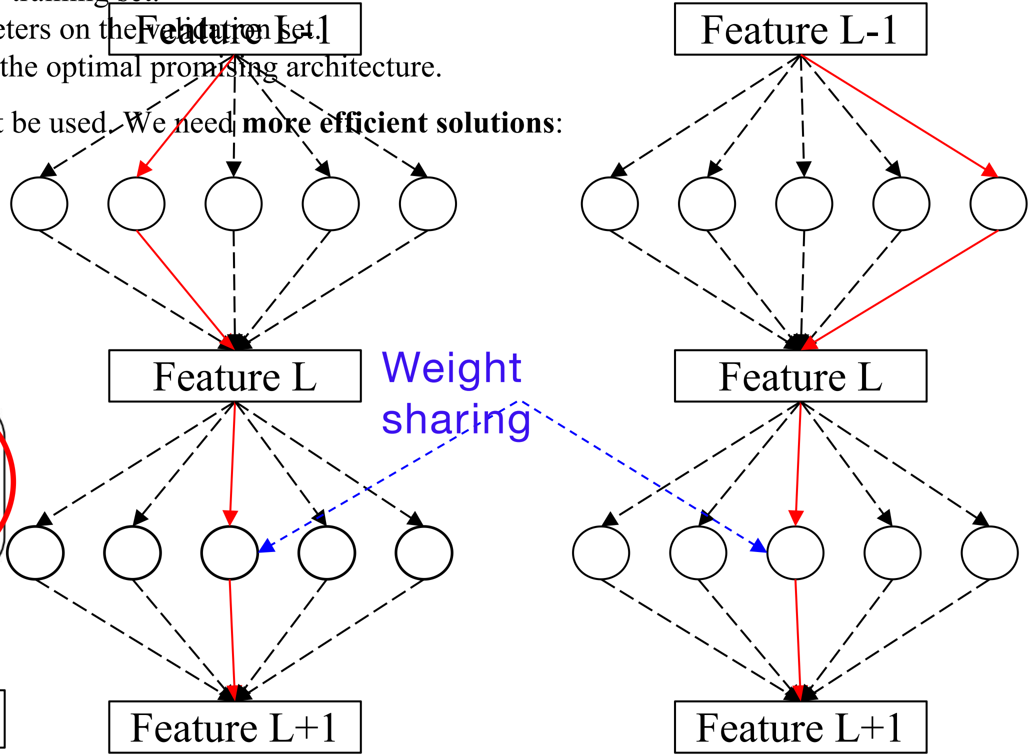
- Step1: Train all possible architectures to convergence on the training set.
- Step2: Evaluate all these networks with fully-trained parameters on the validation set.
- Step3: Rank their performance on the validation set and get the optimal promising architecture.

Due to the unacceptable computational cost, this method cannot be used. We need **more efficient solutions**:

- Replace brute-force exhaustive with **heuristic algorithms**
- Training less epochs or using less data
- **Weight sharing** (one-shot NAS)



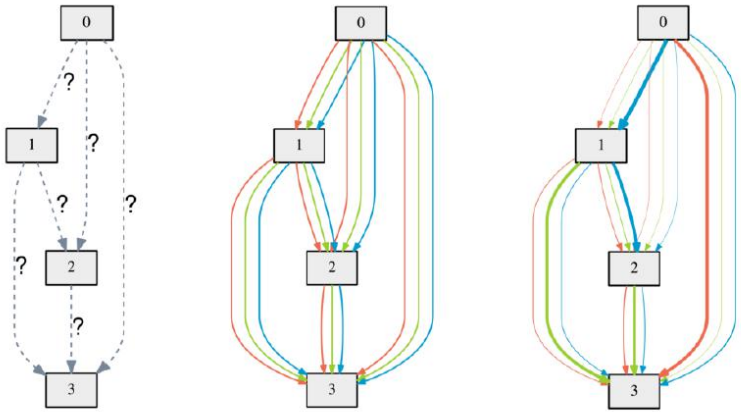
Computation prohibitive on large-scale datasets.



LightTrack

Preliminaries on one-shot NAS

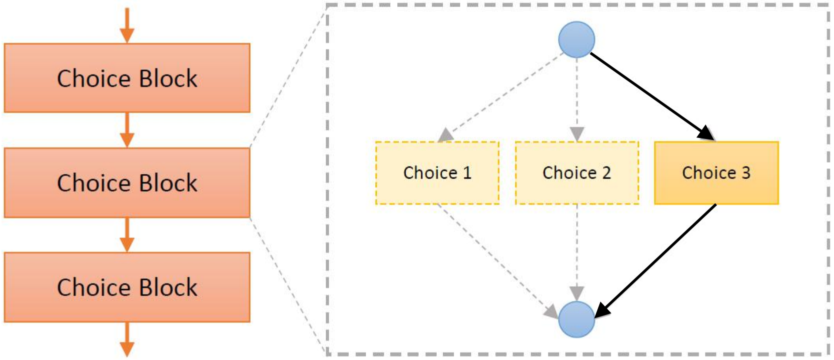
DARTS *v.s* SPOS



(a) (b) (c)

$$\bar{o}^{(i,j)}(x) = \sum_{o\in\mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o'\in\mathcal{O}}\exp(\alpha_{o'}^{(i,j)})} o(x)$$

- Optimize architecture parameters and weight parameters jointly
- Always Keeping all operations in memory



Step1: $W_{\mathcal{A}} = \underset{W}{\operatorname{argmin}} \mathcal{L}_{\text{train}}(\mathcal{N}(\mathcal{A}, W))$

Step2: $a^* = \underset{a\in\mathcal{A}}{\operatorname{argmax}} \text{ACC}_{\text{val}}(\mathcal{N}(a, W_{\mathcal{A}}(a)))$

- Decouple training and searching
- Activate only one path in each iteration

More Resources for Neural Architecture Search

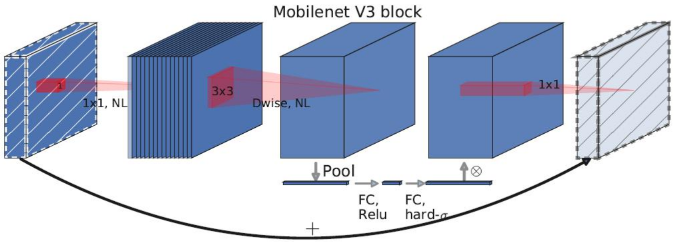
Blogs, Libraries, Benchmarks, and Papers…

- https://github.com/D-X-Y/Awesome-AutoDL
- https://www.automl.org/automl/literature-on-neural-architecture-search/

LightTrack

Method

- Search space



Kernel size: {3x3, 5x5, 7x7}
Expansion ratio: {4, 6}

6 choices for each MBConv

Insights:

- Both **backbone** and **head** play significant roles for a successful tracker
- there is no definitive answer to the question of **which layer's feature** is more suitable for object tracking

Search Space Details

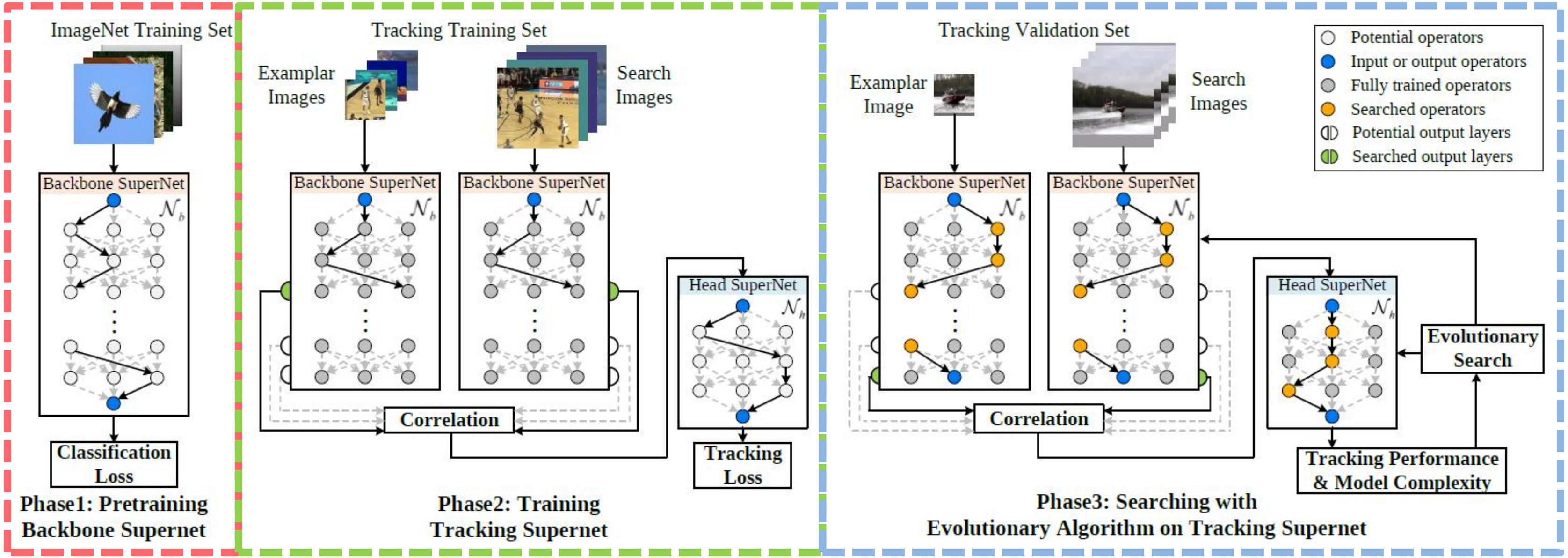
	Input Shape	Operators	$N_{choices}$	Chn	Rpt	Stride
Backbone	$256^2 \times 3$	3×3 Conv	1	16	1	2
	$128^2 \times 16$	DSConv	1	16	1	1
	$128^2 \times 16$	MBConv	6	24	2	2
	$64^2 \times 24$	MBConv	6	40	4	2
	$32^2 \times 40$	MBConv	6	80	4	2
	$16^2 \times 80$	MBConv	6	96	4	1
Cls Head	$16^2 \times 128$	DSConv	6	C_1	1	1
	$16^2 \times C_1$	DSConv / Skip	3	C_1	7	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	7	1
	$16^2 \times C_2$	3x3 Conv	1	4	1	1

DSConv: Depthwise separable convolution
Kernel size {3x3, 5x5}, channel {128, 192, 256}

LightTrack

Method

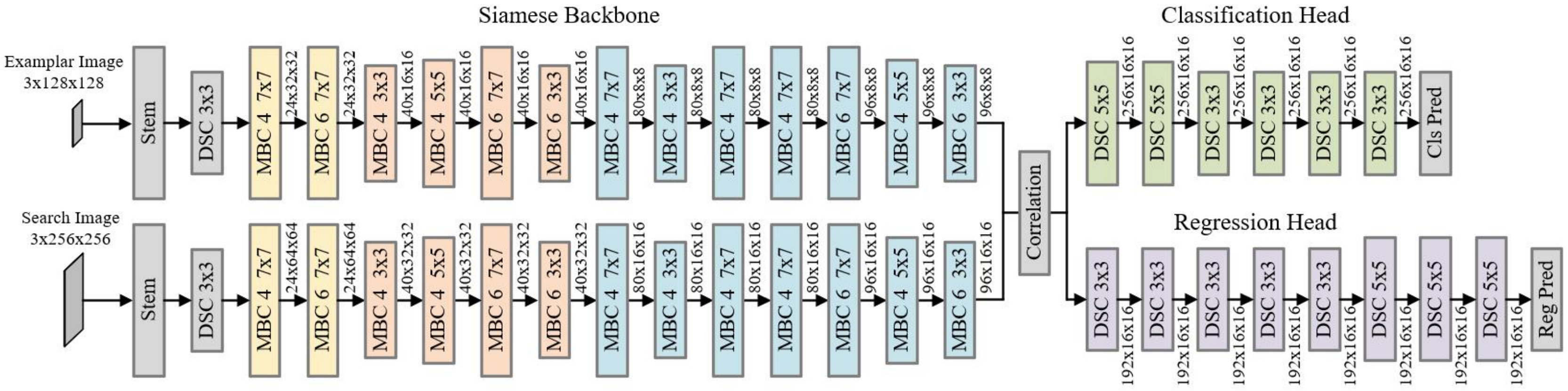
- Framework (pipeline)



LightTrack

Experiments

- Searched architecture



- More than 50% layers in the backbone adopt kernel size 7x7 (large receptive fields can improve the localization precision)
- The searched architecture chooses the second-last block as the feature output layer. (higher-level feature might not be better)

The classification branch contains fewer layers than the regression branch. (coarse object localization is relatively easier than precise bounding box regression.

LightTrack

Experiments

- Comparison with SOTA trackers

VOT2019

	SiamMask [5(	SiamFC++(G)	SiamRPN++(M)	ATOM	TKU	DiMP^r	Ocean(off)	**Ours**	**Ours**	**Ours**
EAO(↑)	0.28									
Accuracy(↑)	0.59									
Robustness(↓)	0.46									
FLOPs(G)(↓)	15.									
Parameters(M)(↓)	16.									

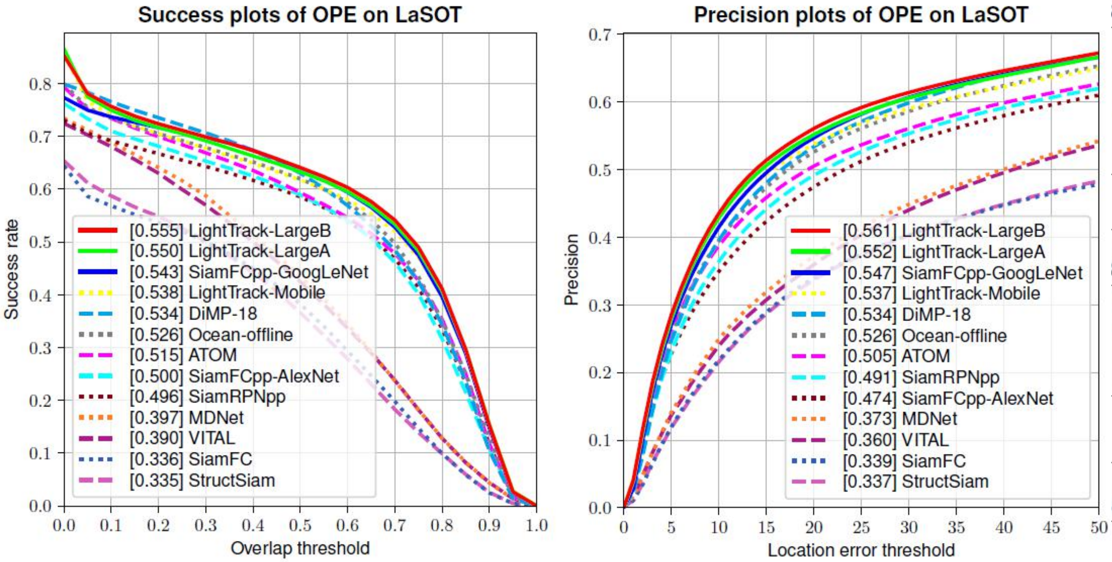
GOT-10K

	DaSiam [57]
AO(↑)	0.417
SR0.5(↑)	0.461
FLOPs(G)(↓)	21.0
Parameters(M)(↓)	19.6

TrackingNet

	RTMDNet [27]										
P(%)	53.3										
P_{norm}(%)	69.4										
AUC(%)	58.4	61.2	63.8	66.9	70.3	71.2	73.3	74.0	72.5	73.6	73.3

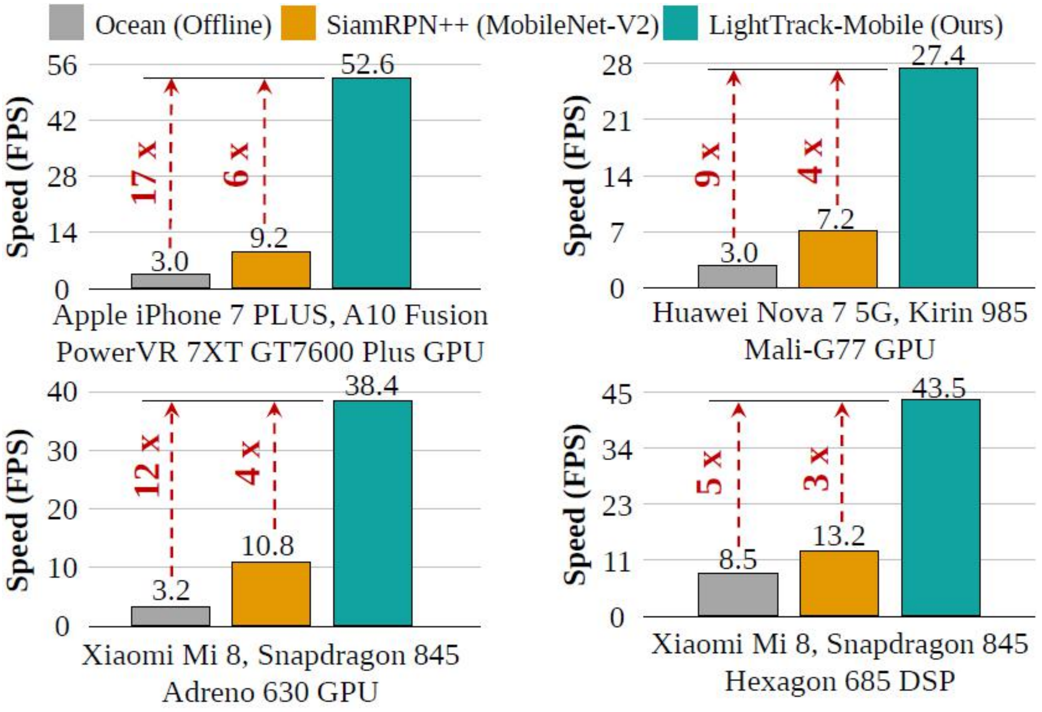
LaSOT



LightTrack

Experiments

- Speed on resource-limited platforms



Thanks for your listening

We are hiring!
Please contact to Houwen.peng@microsoft.com