Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning

Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Jin Young Choi

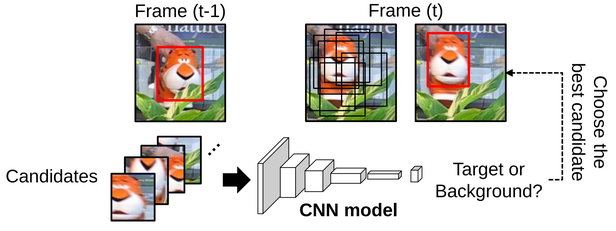
SEOUL NATIONAL UNIVERSITY

CVPR July 21-26 HONOLULU 2017

Motivation

Visual tracking

- Find the target position in a new frame.
- Deep CNN-based tracking method (Tracking-by-detection)



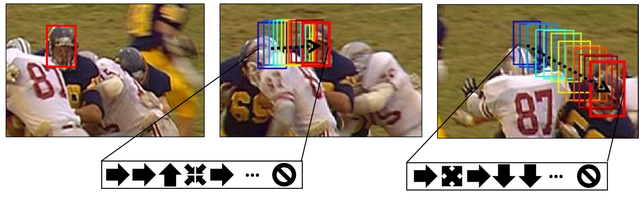
Problem

- Inefficient search strategy.
- Need lots of labeled video frames to train CNNs.

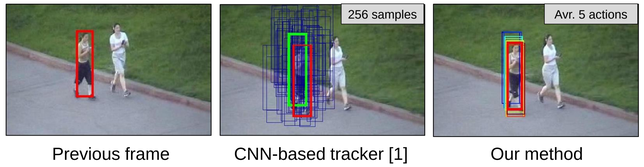
Approach

Action-driven tracking

- Dynamically capture the target by selecting sequential actions.



- Comparison with existing method.

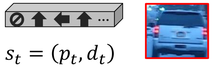


Previous frame | CNN-based tracker [1] | Our method

[1] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking., CVPR 2016.

Action-Decision Networks (ADNet)

Problem setting (Markov decision process)



- Action: a_t, defined by discrete actions

Translation moves | Scale changes | Stop

- State: $s_t = (p_t, d_t)$

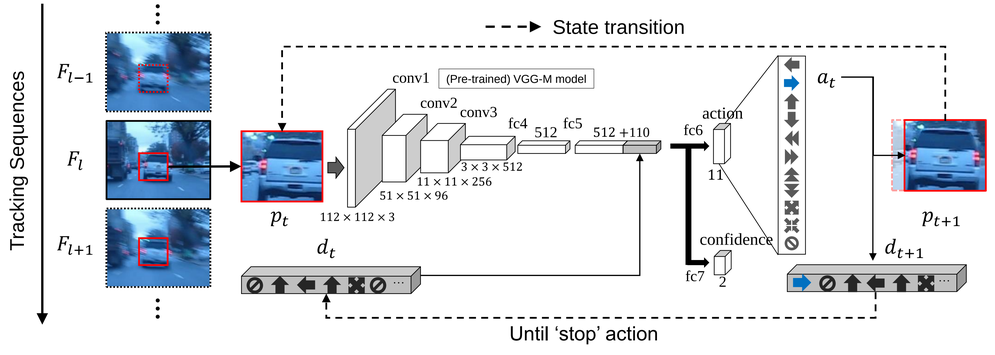
Image patch: $p_t \in \mathbb{R}^{112*112*3}$

Action dynamics: $d_t \in \mathbb{R}^{110}$

- Reward:

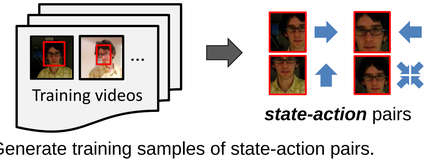
$$r(s_t) = \begin{cases} 1, \text{if } IOU(p_t, G) > 0.7 \\ -1, \text{otherwise} \end{cases}$$

Action-Decision Network



Training method

- Supervised learning



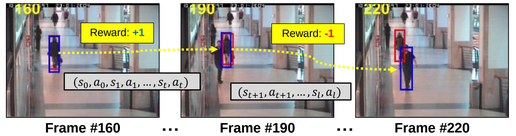
state-action pairs

- Generate training samples of state-action pairs.
- Train policy network as multiclass classification with softmax.
- Then, reinforcement learning is performed to improve ADNet.

Online Tracking

- Initial fine-tuning with N_I initial samples.
- Online adaptation on every I frames with N_O online samples.
- Re-detection is conducted when the target is missed.

- **Reinforcement learning**



Frame #160 ··· Frame #190 ··· Frame #220

- Can handle the *semi-supervised* case.
- Run tracking to a piece of training sequence.
- Get trajectory of states, actions, and rewards $\{(s_t, a_t, r_t)\}$.
- By REINFORCE algorithm,

$$\Delta W \propto \sum_t^T \frac{\partial \log\left(p(a_t|s_t; W)\right)}{\partial W} r_t$$

- Train policy network to maximize the expected tracking rewards.

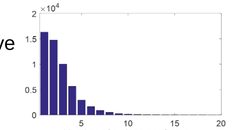
Experiments

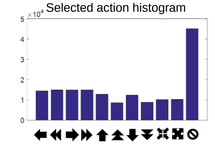
Experiment setting

- MatConvNet toolbox, i7-4790K CPU, Nvidia Titan X GPU.
- Trained on VOT dataset & evaluated on OTB-100 dataset.
- ADNet (N_I: 3000, N_O: 250, I: 10) → 3 fps (Prec: 88%)
- ADNet-fast (N_I: 300, N_O: 50, I: 30) → 15 fps (Prec: 85%)

Analysis on action

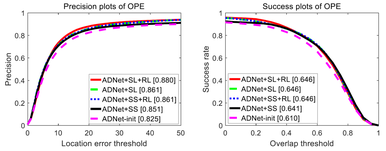
- 93% of total frames have smaller than 5 actions





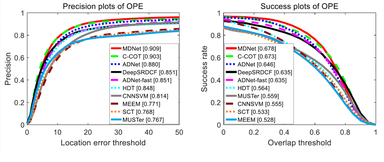
Self-comparison

- init: no pre-training, only online adaptation
- SL: supervised learning
- SS: uses 1/10 gt annotations
- RL: reinforcement learning



OTB-100 test results

	Algorithm	Prec.(20px)	IOU(AUC)	FPS	GPU
	ADNet	88.0%	0.646	2.9	O
	ADNet-fast	85.1%	0.635	15.0	O
Non real-time	MDNet [24]	90.9%	0.678	< 1	O
	C-COT [9]	90.3%	0.673	< 1	O
	DeepSRDCF [8]	85.1%	0.635	< 1	O
	HDT [25]	84.8%	0.564	5.8	O
	MUSTer [28]	76.7%	0.528	3.9	X
Real-time	MEEM [42]	77.1%	0.528	19.5	X
	SCT [3]	76.8%	0.533	40.0	X
	KCF [13]	69.7%	0.479	223	X
	DSST [7]	69.3%	0.520	25.4	X
	GOTURN [12]	56.5%	0.425	125	O



Sequential actions selected by ADNet