The Visual Object Tracking Challenge Workshop 2016

Modeling and Propagating CNNs in a Tree Structure for Visual Tracking

Hyeonseob Nam*   Mooyeol Baek*   Bohyung Han

ComputerVision Lab.

Dept. of Computer Science and Engineering

POSTECH

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

*Equal contributors

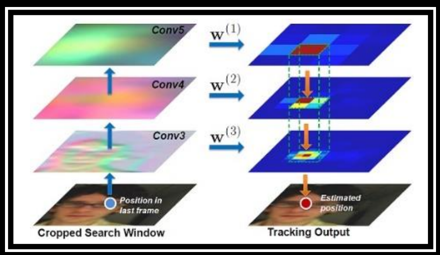
Key concept

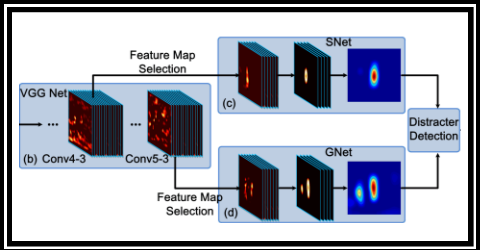
Multiple CNNs trained on
tracking results from different time slices

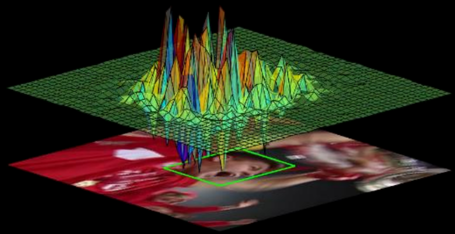
Modeling and propagating
CNNs in **a tree structure**

Deep Learning for Visual Tracking

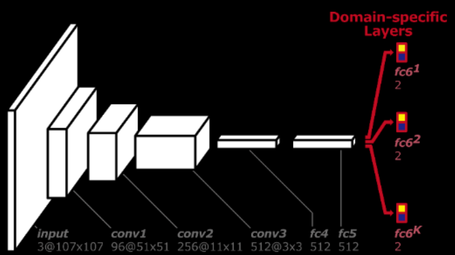
- Response map from CNN features
  - HCF[Ma2016] FCNT[Wang2015] DeepSRDCF[Danelljan2015]







- Pre-train CNN with tracking datasets
  - MDNet[Nam2015]



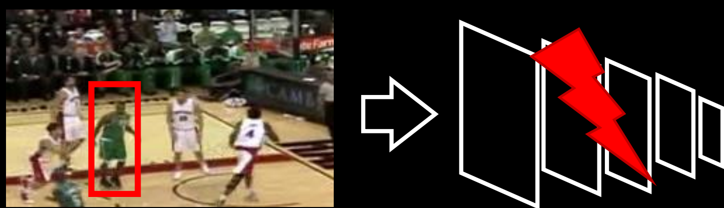
Comparable with other shallow-feature trackers.

Uses only a few layer of CNN! – Not DEEP

Needs much number of tracking videos to train.

Challenges

- Less training data

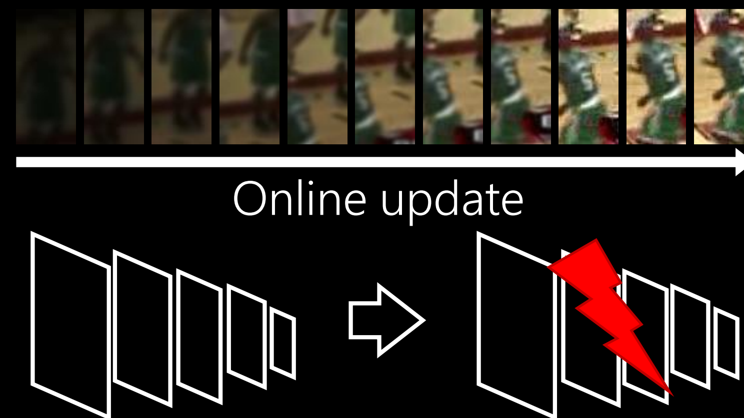


Only a single labeled frame

Overfitting

Difficult to train

- Catastrophic forgetting



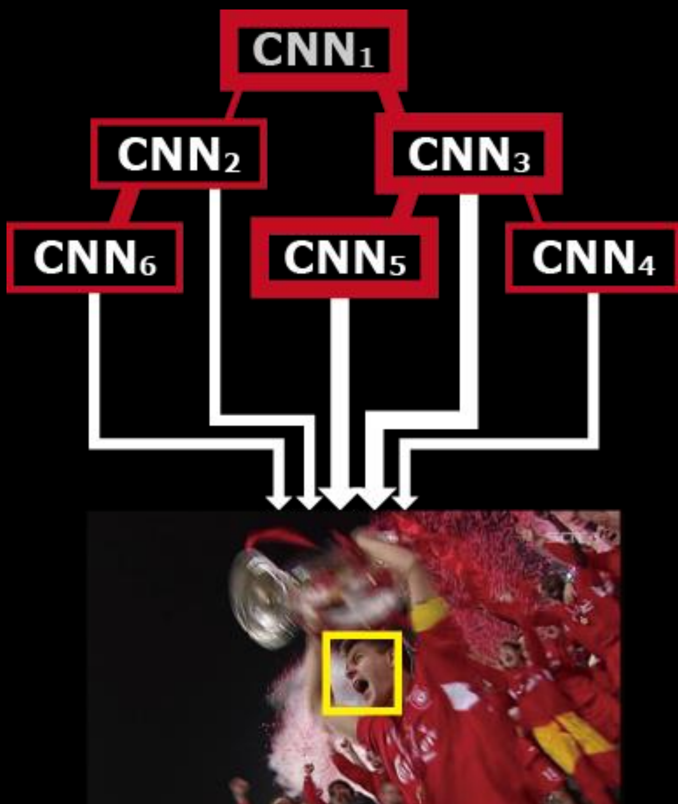
CNN “forgets” previous information

Unstable model

Easily drifts

Our method

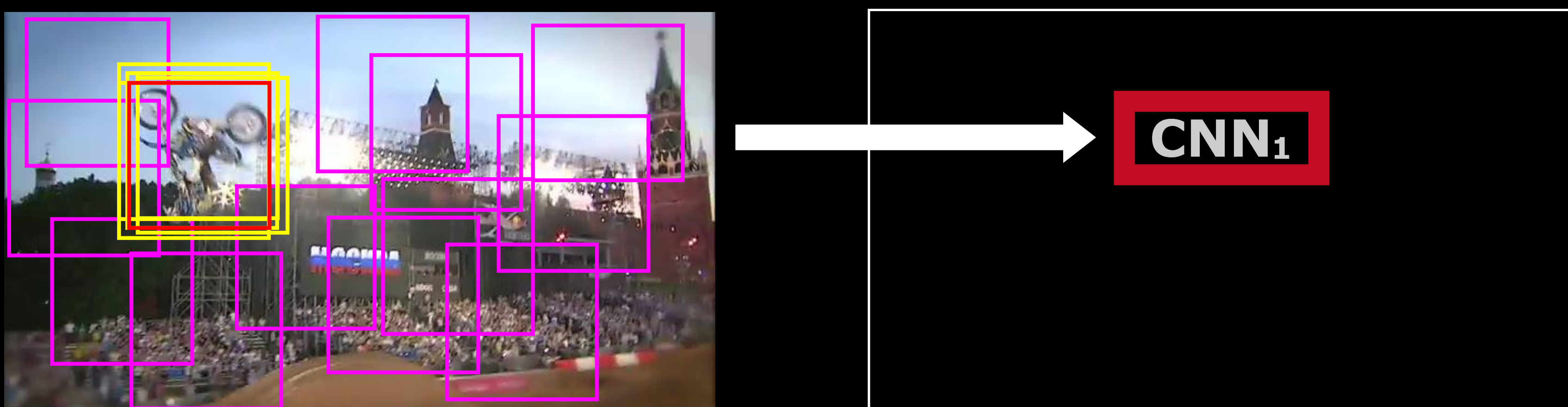
- CNNs trained on frames from different time slices
- State estimation



- Model update

Overall Algorithm – 1

- Initialize model with ground-truth



1st frame

Model

ground-truth

positive patches

negative patches

Overall Algorithm – 2

- Estimate target location for T consecutive frames

(T+1)st frame

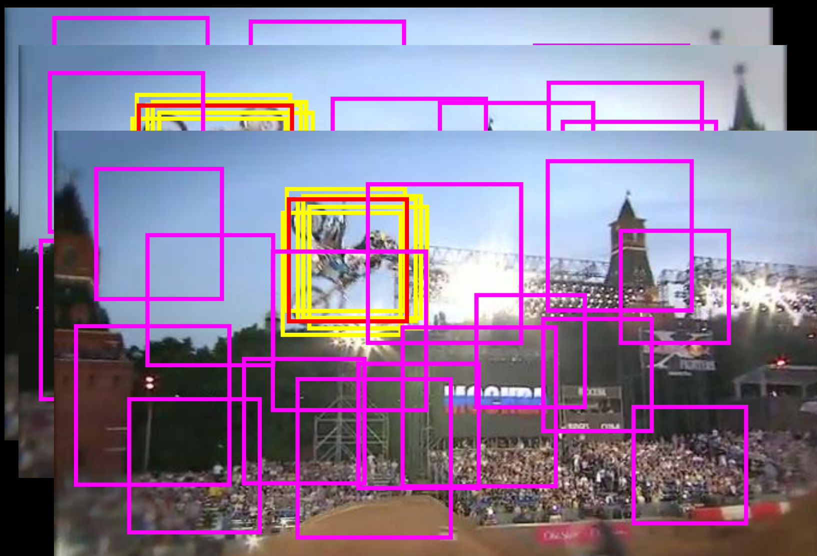
Model

tracking results

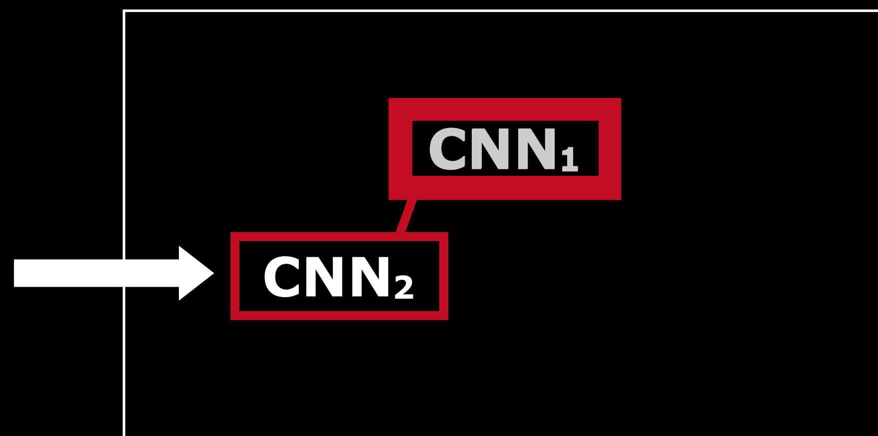
target candidates

Overall Algorithm – 3

- Update model



(T+1)st frame



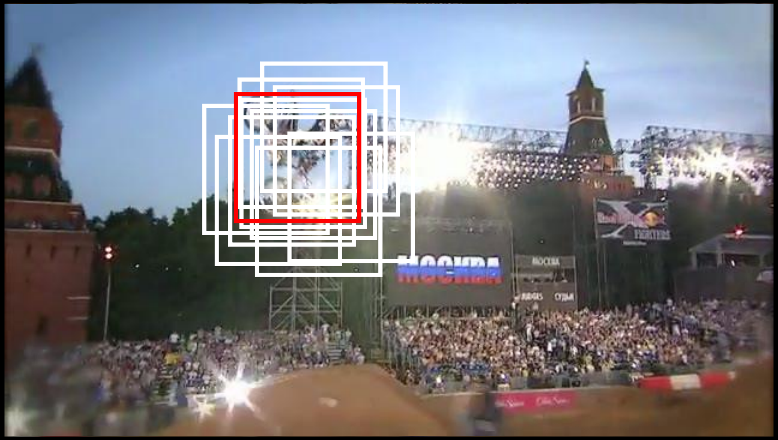
Model

tracking results
positive patches
negative patches

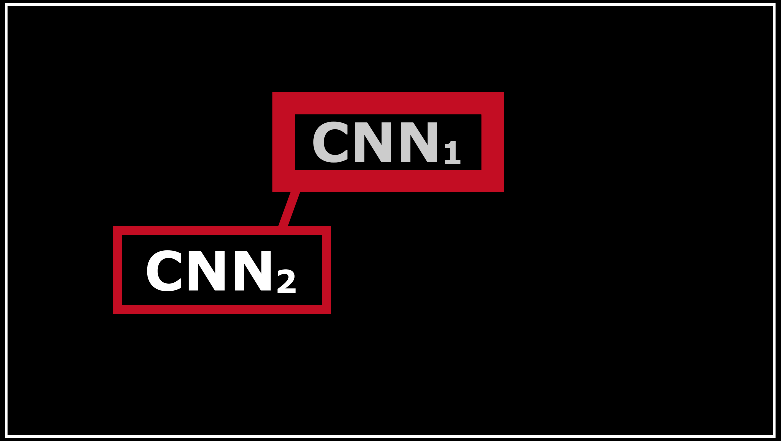
- Choose CNN with the best target score as a parent
- Maintain **N** latest CNNs

Overall Algorithm – 4

- Iterate until the end of sequence



$(T+2)^{nd}$ frame

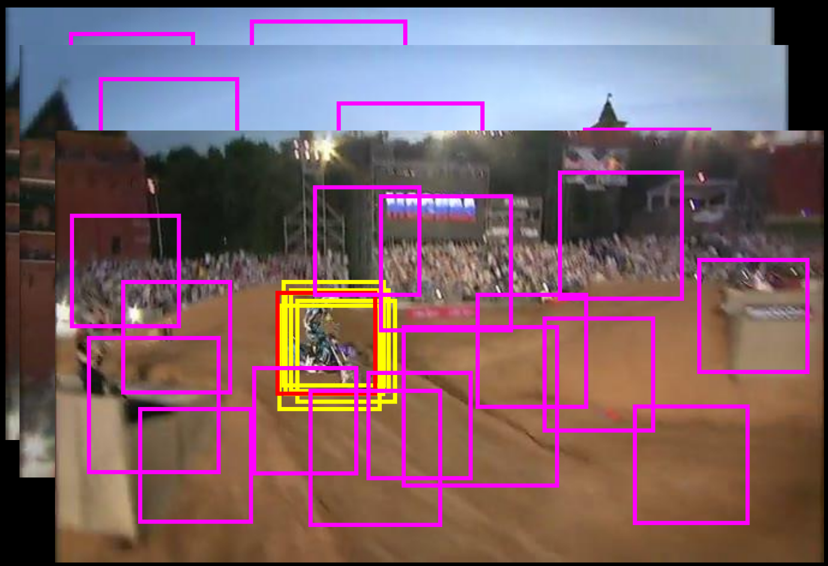


Model

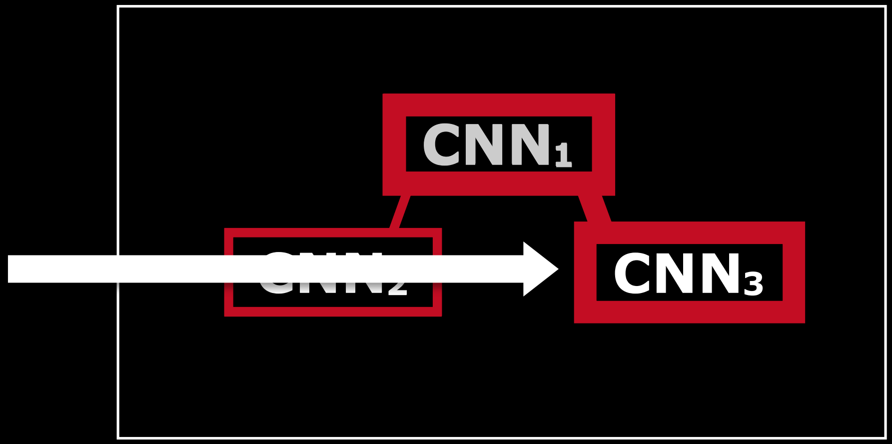
tracking results
target candidates

Overall Algorithm – 4

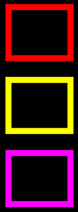
- Iterate until the end of sequence



(2T+1)st frame

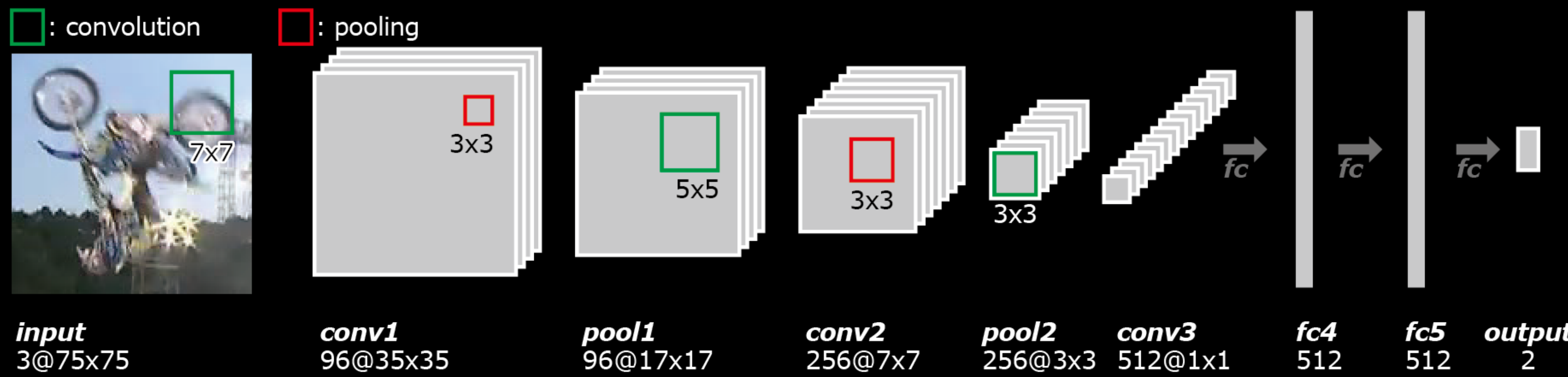


Model



- Choose CNN with the best target score as a parent
- Maintain **N** latest CNNs

Structure of CNN

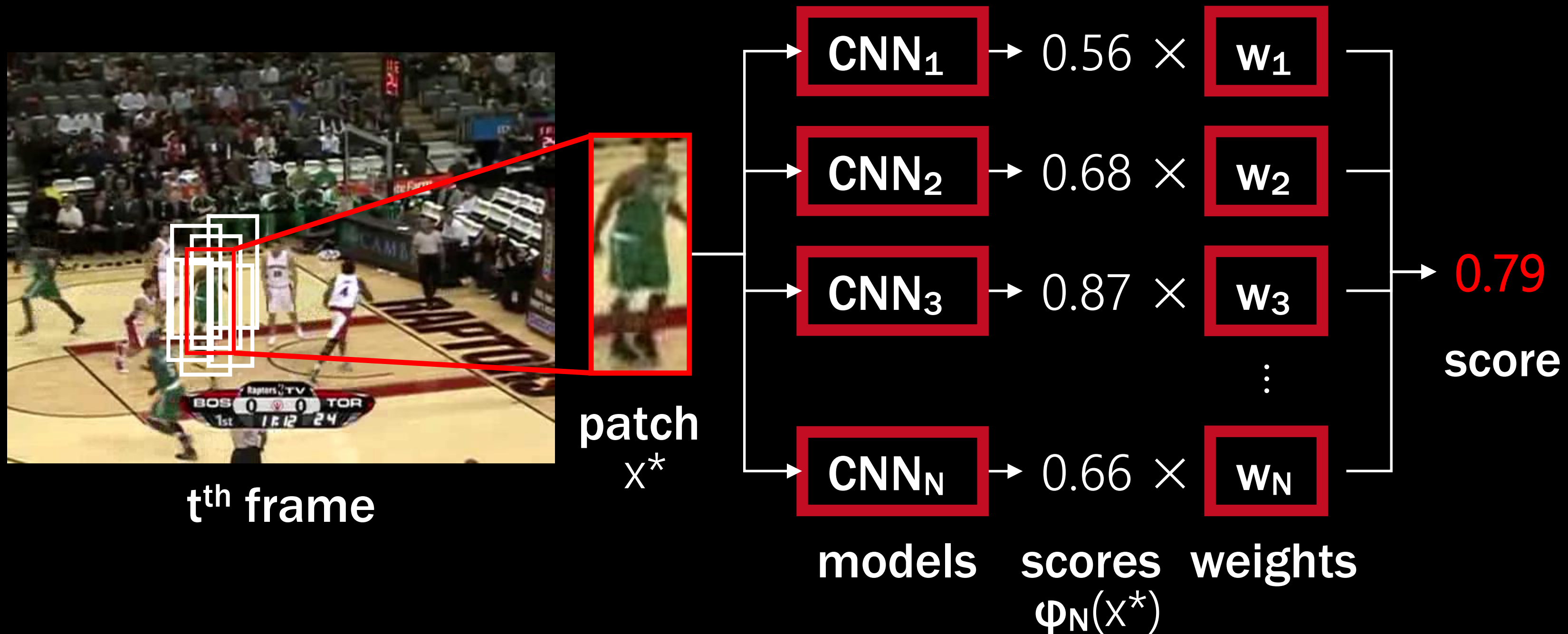


- Convolution layers are from VGG-M[Chatfield2014] network.
- Output: normalized vector $[\boldsymbol{\varphi}(x), 1-\boldsymbol{\varphi}(x)]^T$
- Fully-connected layers are randomly initialized.
- Update fully-connected layers only (**fc4, fc5, output**)

State Estimation

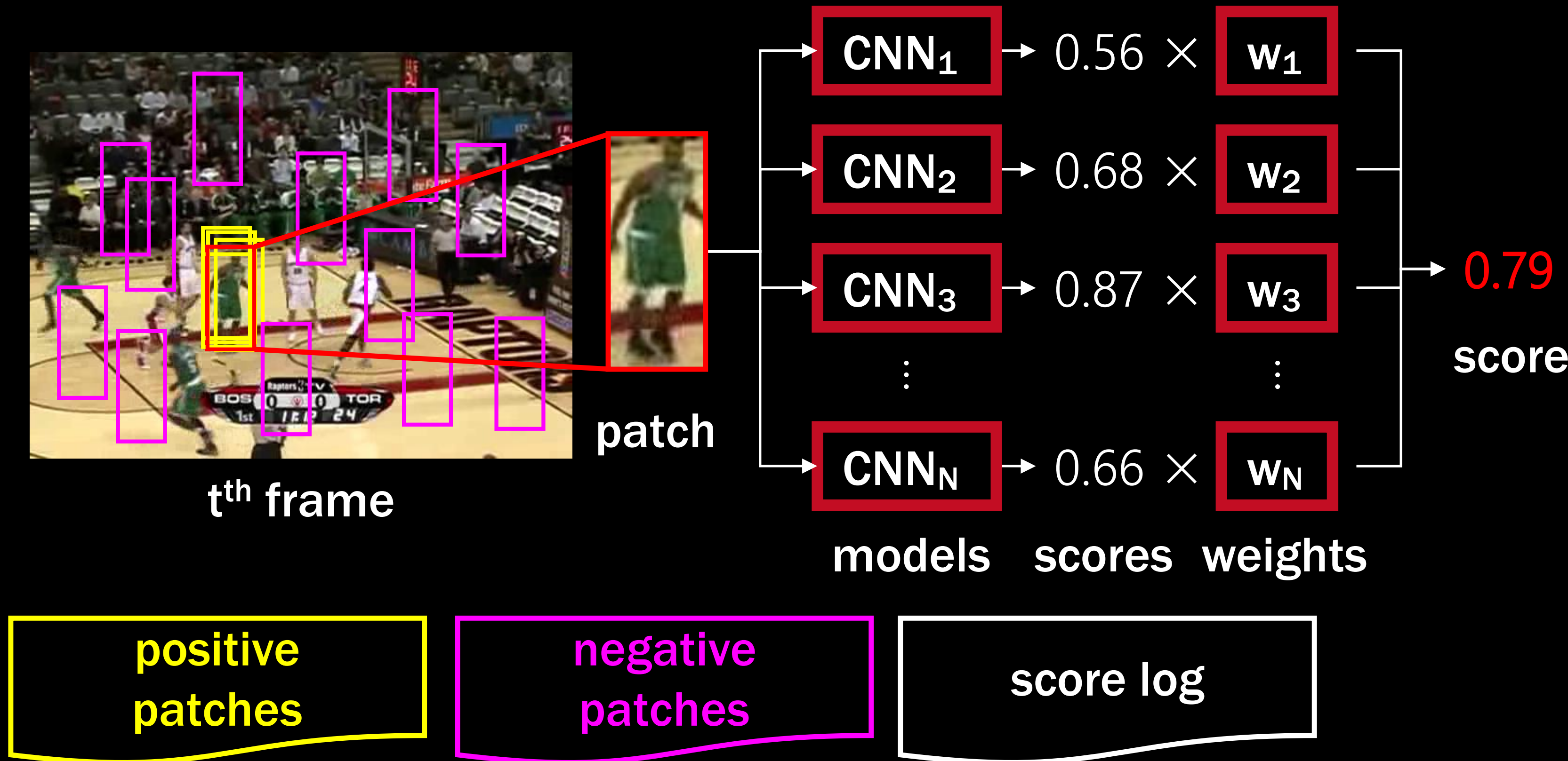
- **Weighted-average** scores from multiple CNNs
- $\boxed{w_N}$: $\min(\max(\varphi_N(x)), \beta_N)$

affinity of CNN_N with t^{th} frame reliability of CNN_N



Model Update – 1

- Sample **training data** for model update
- Store **target scores** of each CNN for parent selection

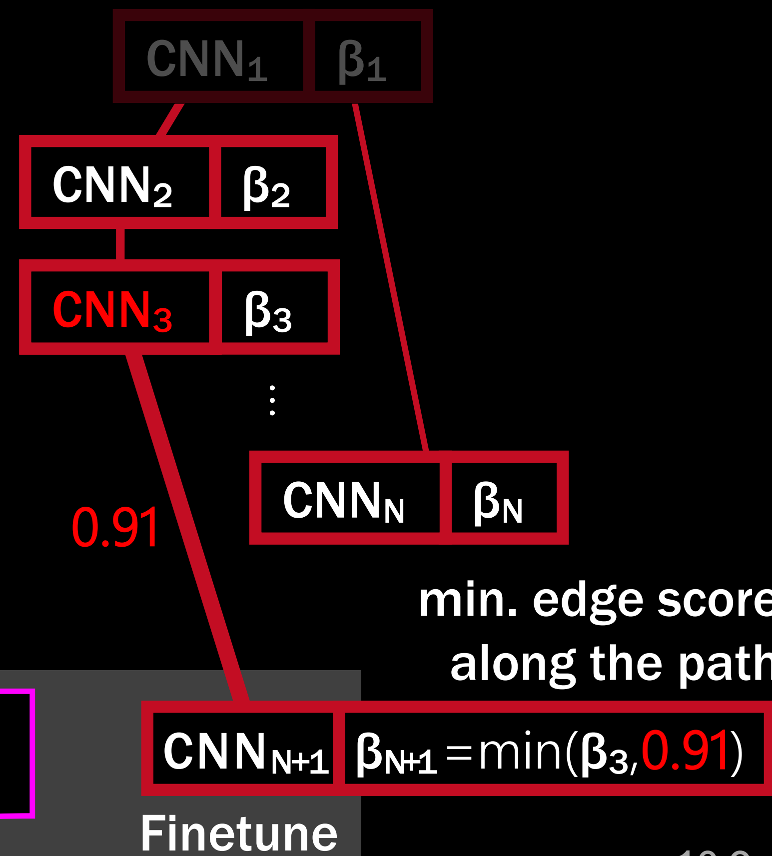


Model Update – 2

Score log

Model	Target scores t+1	t+2	⋯	t+T		Avg.
CNN_1	0.56	0.61		0.54	→	0.53
CNN_2	0.68	0.71	⋯	0.82	→	0.79
CNN_3	0.87	0.83		0.93	→	max 0.91
⋮		⋮				⋮
CNN_N	0.66	0.56	⋯	0.62	→	0.68

Disable old model
(maintain N latest models)

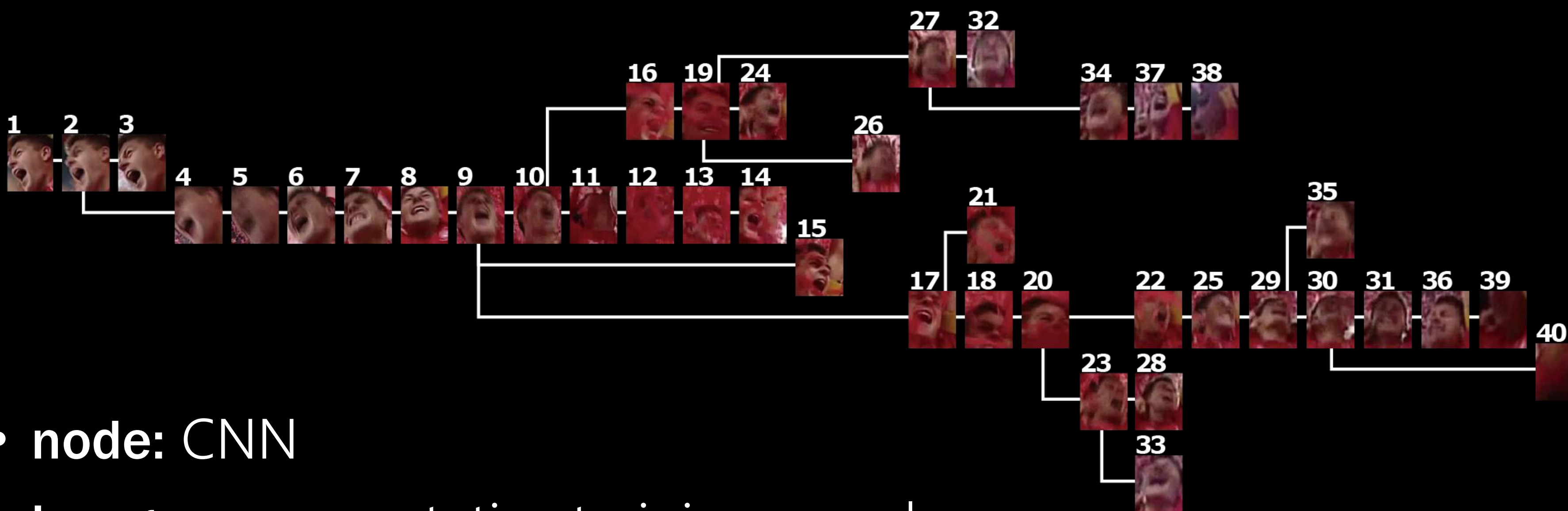


positive patches

negative patches

Model Update – an Example

- *Soccer* sequence



- **node:** CNN
- **image:** representative training sample

- Multiple CNN captures **multi-modal appearances**
- Tree structure **isolates erroneous models** in a branch

Experiment

- Dataset
  - VOT2016, VOT2015[Kristan2015], OTB50[Wu2013]
- Experiment settings
  - Parameters fixed throughout the whole experiment
- Speed: **1.5 fps**
  - Intel Core i7 3.3Ghz with NVIDIA TITAN X
  - MATLAB / MatConvNet

Results – VOT2016

Overlap: 0.56, Failures: 0.83, Unsupervised overlap: 0.49

Red box: tracking result, Black box: ground-truth

Results – VOT2015[Kristan2015]

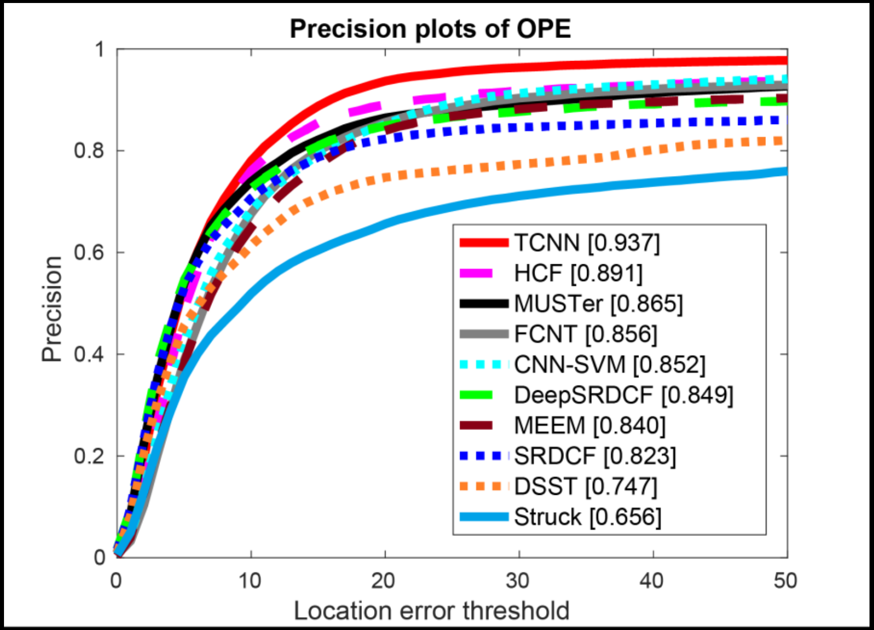
Trackers	Accuracy		Robustness		Expected overlap ratio
	Rank	Score	Rank	Score	
DSST[Danelljan2014]	3.48	0.54	7.93	2.56	0.1719
MUSTer[Hong2015CVPR]	3.42	0.52	6.13	2.00	0.1948
MEEM[Zhang2014]	4.08	0.50	6.02	1.85	0.2212
Struck[Hare2011]	5.27	0.47	4.05	1.64	0.2389
RAJSSC[Zhang2015]	2.08	0.57	4.87	1.63	0.2420
NSAMF[Li2014]	3.22	0.53	3.85	1.29	0.2536
SC-EBT[Wang2014]	2.27	0.55	5.07	1.90	0.2540
sPST[Hua2015]	2.78	0.55	4.67	1.48	0.2767
LDP[Lukežič2016]	4.58	0.49	4.65	1.33	0.2785
SRDCF[Danelljan2015ICCV]	2.32	0.56	3.48	1.24	0.2877
EBT[Zhu2016]	6.30	0.48	2.75	1.02	0.3160
DeepSRDCF[Danelljan2015ICCVW]	2.23	0.56	2.90	1.05	0.3181
TCNN (ours)	1.58	0.59	2.83	0.74	0.3404

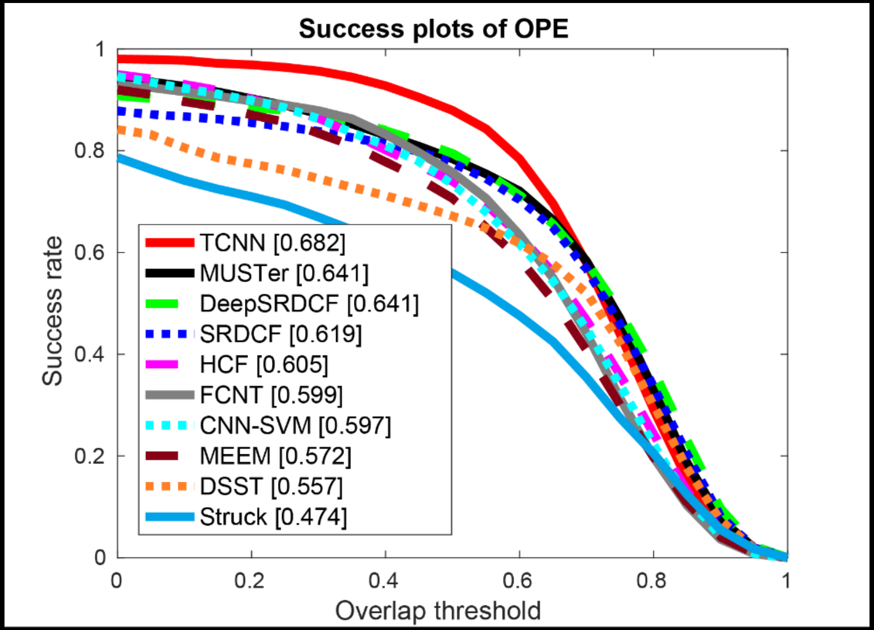
*MDNet excluded

Red: best, Yellow: second best

Results – OTB50[Wu2013]

- TCNN (ours) compared to state-of-the-art trackers
  - HCF[Ma2016], CNN-SVM[Hong2015ICML], MEEM[Zhang2014], SRDCF[Danelljan2015ICCV], DeepSRDCF[Danelljan2015ICCVW], MUSTer[Hong2015CVPR], FCNT[Wang2015], DSST[Danelljan2014], Struck[Hare2011]





Ablation study

- Single
  - Single CNN
- Linear
  - Models propagating in a linear chain
- Tree-mean
  - Models propagating in a tree structure
  - Equal weights for every model
- **Tree-weighted**
  - Models and weights propagating in a tree structure

- Results on OTB50

Variations	Precision	Success	AUC
Single	0.896	0.858	0.658
Linear	0.920	0.869	0.672
Tree-mean	0.928	0.868	0.674
Tree-weighted	0.937	0.879	0.682

Comparison with MDNet

- Results on VOT2016

Trackers	Accuracy	Robustness
MDNet-N	0.54	0.91
TCNN (ours)	0.56	0.83
MDNet	0.57	0.76

- Results on OTB50

Trackers	Precision	Success	AUC
MDNet-N	0.896	0.858	0.658
TCNN (ours)	0.920	0.869	0.672
MDNet	0.928	0.868	0.674

- MDNet-N: MDNet without pretraining using tracking videos

Conclusion

- Multiple CNNs are helpful to achieve **multi-modality and reliability** of target appearances.
- The CNNs and their weights propagating in a tree structure efficiently **isolates error** in a temporal manner.

Please refer to our arXiv paper for further details.
https://arxiv.org/abs/1608.07242

Thank you.