鲁棒视觉目标跟踪的深度攻击问题研究

马超

2022-1-8

Research interests: computer vision

人工智能研究院
Artificial Intelligence Institute

Motion and Tracking

CVPR15 (IJCV18), ICCV15 (TPAMI20), ICCV17, CVPR18, ECCV18, NeurIPS18, CVPR19, CVPR19, CVPR19, CVPR19, ECCV20 , CVPR 21, ICCV21

Q: What is the dark green vegetable?
A: Cucumber

Vision and Language

CVPR18, ECCV20

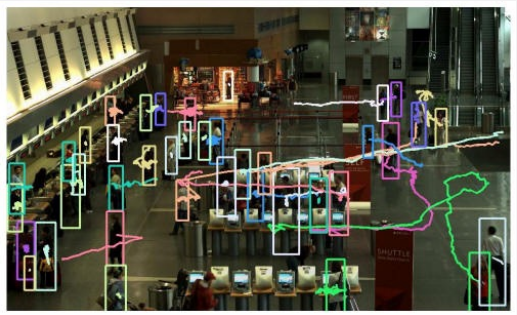
Image Restoration

ECCV14, CVIU17, ECCV20

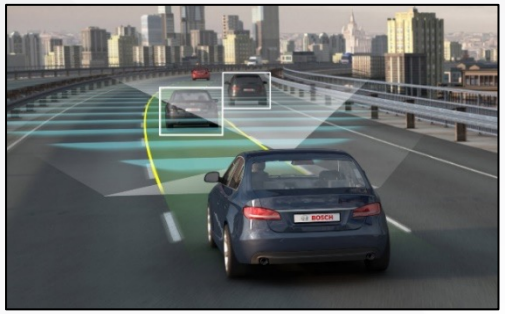
3D Reconstruction

CVPR20 Workshop Best Paper

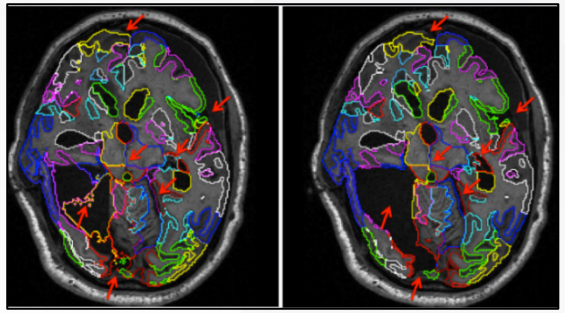
Visual Object Tracking and Applications



Intelligent Surveillance



Autonomous Driving



Medical Imaging

Human-Computer Interaction

Images from Google Search

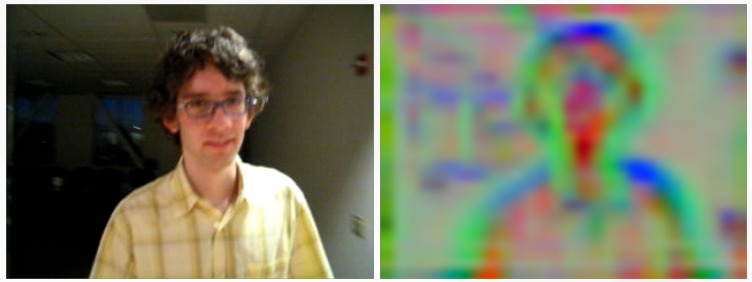
Problems in Deep Tracking

LCT CVPR15 IJCV18
HCF ICCV15 TPAMI18
CREST ICCV17
DSLT ECCV18
TADT CVPR19 IJCV20
UDT CVPR19
COSNet CVPR19
3D Tracking CVPR21, IROS21

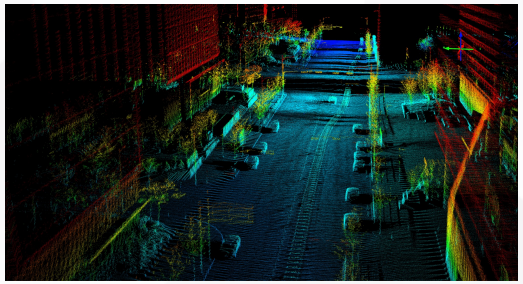
2015 2017 2018 2019 2020 2021

Shallow ← → Deep

VITAL CVPR18
DAT NeurIPS18
DAIN CVPR19
RobTracker ECCV20



Representation Learning for Tracking

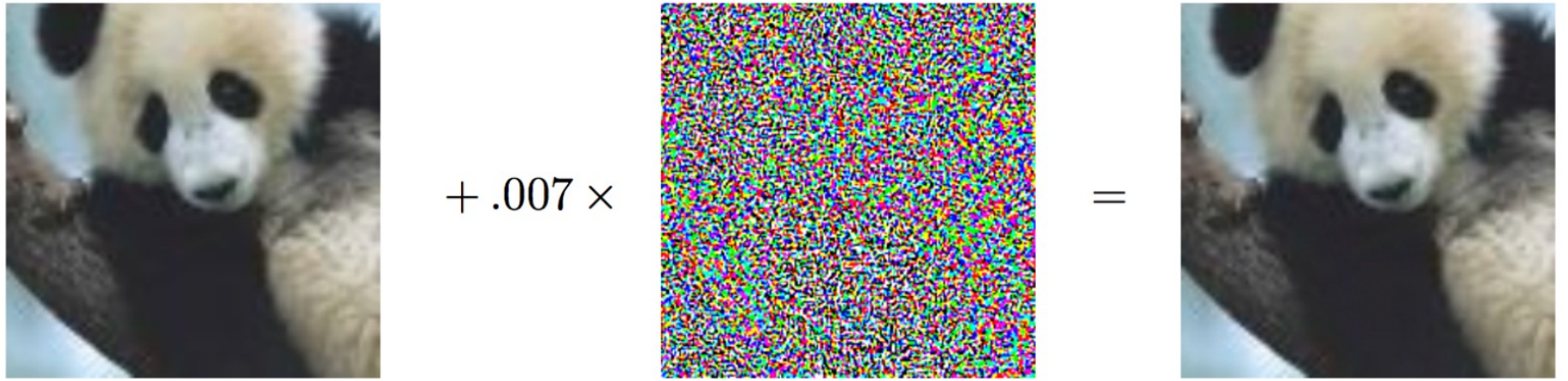


Cross-Modality Tracking

Robust Object Tracking

Introduction

Clean image/ Adversarial input **Perturbation** **Adversarial example**



- Adversarial examples are inputs to machine learning models that an attacker intentionally designed to cause the model to make mistakes.
- Threat model defines the rules of the attack.

[1] Goodfellow I J, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. In: ICLR (2015)

Taxonomy

Attack:

- Targeted Attack / Non-targeted Attack;
- Digital attack / physical attack;
- Single-step attack / iterative attack;
- White-box attack: The adversary has access to all the information of the target neural network.
- Black-box attack: The inner configuration of DNN models is unavailable to adversaries.
- Transfer-based, Score-based and Decision-based attacks.

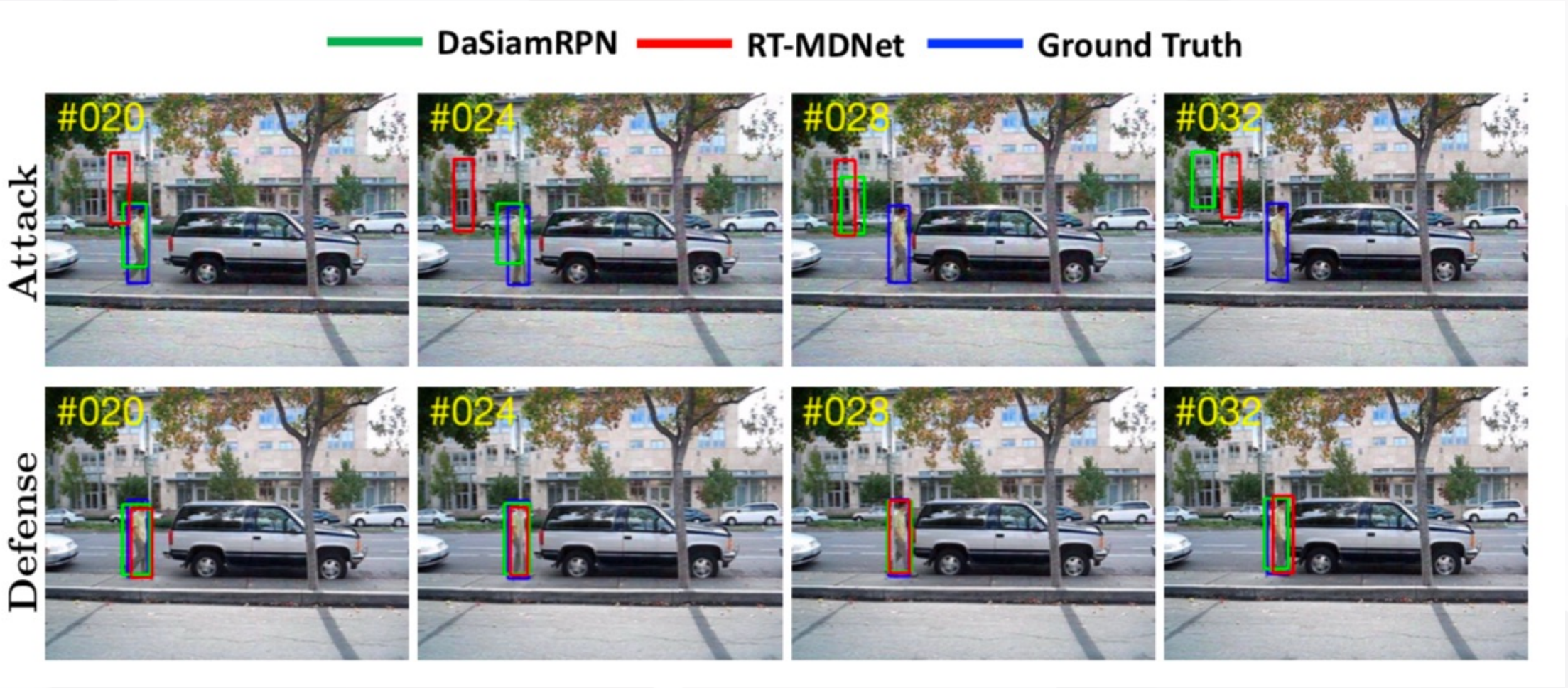
Defense:

- Gradient Masking, Robust Optimization and Adversary Detection.

目　录

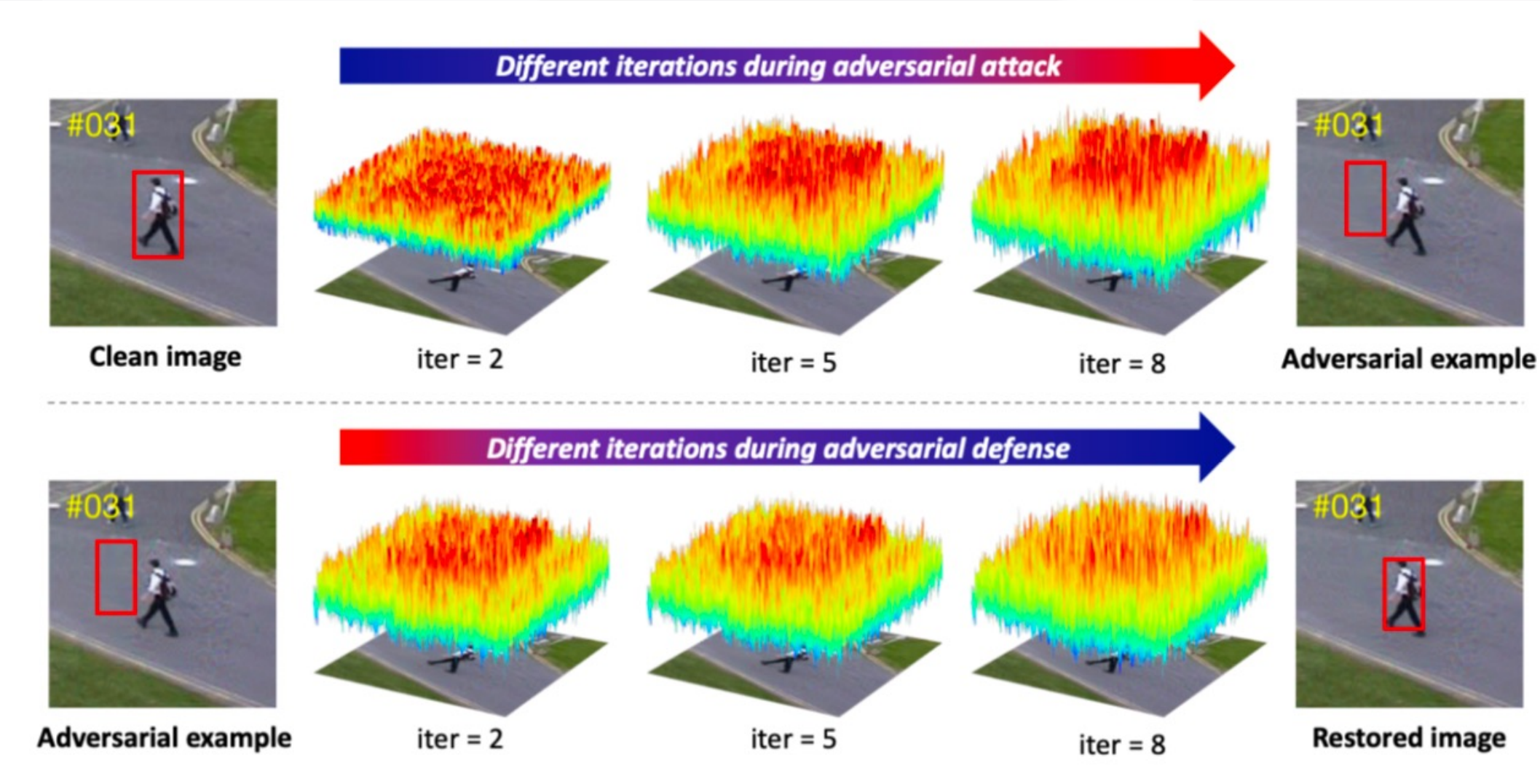
- 视觉跟踪算法白盒攻击（ECCV 2020）
- 目标跟踪算法黑盒攻击（CVPR 2021）

Introduction



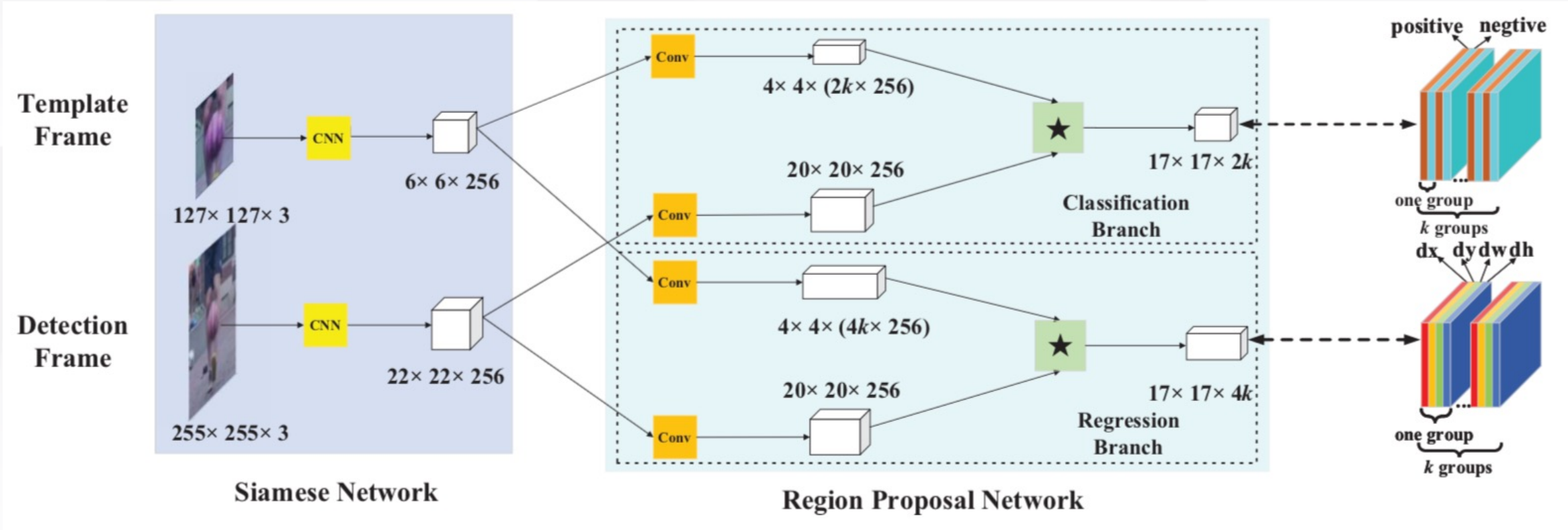
Adversarial examples for attack and defend on the *David3* sequence from OTB2015 dataset

Our Motivations



Variations of adversarial perturbations during attack and defense.

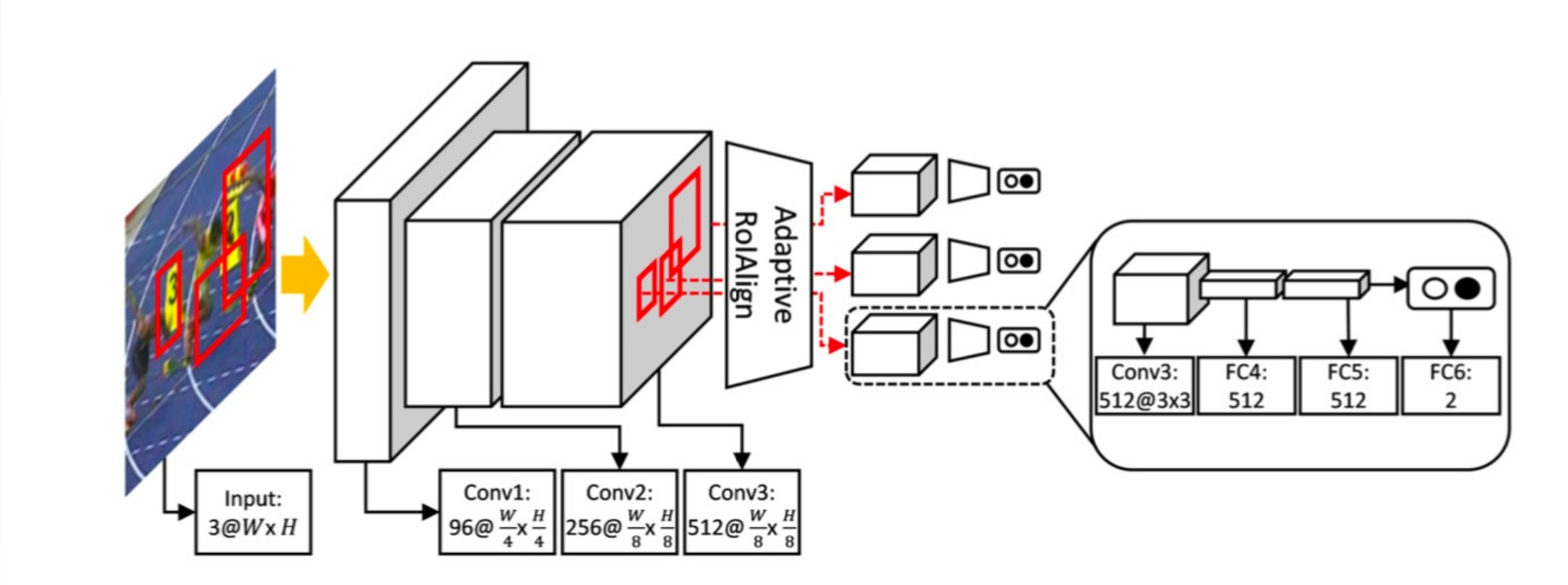
Baseline Tracker 1: DaSiamRPN



- DaSiamRPN is a end-to-end trained off-line tracker, consisting of Siamese subnetwork for feature extraction and region proposal subnetwork including the classification branch and regression branch.

[1] Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware siamese networks for visual object tracking. In: ECCV (2018)

Baseline Tracker 2: RT-MDNet



- RT-MDNet is composed of shared layers and multiple branches of domain-specific layers. When tracking a target in a new sequence, it combines the shared with a new binary classification layer, which is updated online.

[1] Jung, I., Son, J., Baek, M., Han, B.: Real-time mdnet. In: ECCV (2018)

Our Method: Adversarial Example Generation

人工智能研究院
Artificial Intelligence Institute

Algorithm 1: Adversarial Example Generation

Input: input video V with T frames;
target location S^1;
Output: adversarial examples of T frames;

for $t = 2$ **to** T **do**
  Get current frame I_1^t;
  if $t \neq 2$ **then**
    Update I_1^t via Eq. 6;
  end
  for $m = 1$ **to** M **do**
    Create p_c and p_r via IoU ratios between proposals and target location S^{t-1};
    Create $p_c^\star$ by reversing elements of p_c;
    Create $p_r^\star$ via Eq. 3;
    Generate adversarial loss via Eq. 2;
    Update I_m^t via Eq. 5;
  end
  return I_M^t;
end

Temporal attack

$$I_1^t = I_1^t + (I_M^{t-1} - I_1^{t-1})$$

$$\begin{aligned} x_r^\star &= x_r + \delta_{\text{offset}} \\ y_r^\star &= y_r + \delta_{\text{offset}} \\ w_r^\star &= w_r * \delta_{\text{scale}} \\ h_r^\star &= h_r * \delta_{\text{scale}} \end{aligned}$$

$$\begin{aligned} \mathcal{L}_{adv}(I, N, \theta) = \sum_{n=1}^{N} \{ & [L_c(I_n, p_c, \theta) - L_c(I_n, p_c^\star, \theta)] \\ & + \lambda \cdot [L_r(I_n, p_r, \theta) - L_r(I_n, p_r^\star, \theta)] \} \end{aligned}$$

$$I_{m+1} = I_m + \alpha \cdot sign(r_m)$$

Our Method: Adversarial Example Defense

Algorithm 2: Adversarial Example Defense

Input: input video V with T adversarial examples; target location S^1;

Output: adversarial examples of T frames;

for $t = 2$ **to** T **do**
  Get current frame I_1^t;
  if $t \neq 2$ **then**
    Update I_1^t via Eq. 8;
  end
  for $m = 1$ **to** M **do**
    Create p_c and p_r via IoU ratios between proposals and target location S^{t-1};
    Create $p_c^\star$ by reversing elements of p_c;
    Create $p_r^\star$ via Eq. 3;
    Generate adversarial loss via Eq. 2;
    Update I_m^t via Eq. 7;
  end
  return I_M^t;
end

Temporal defense

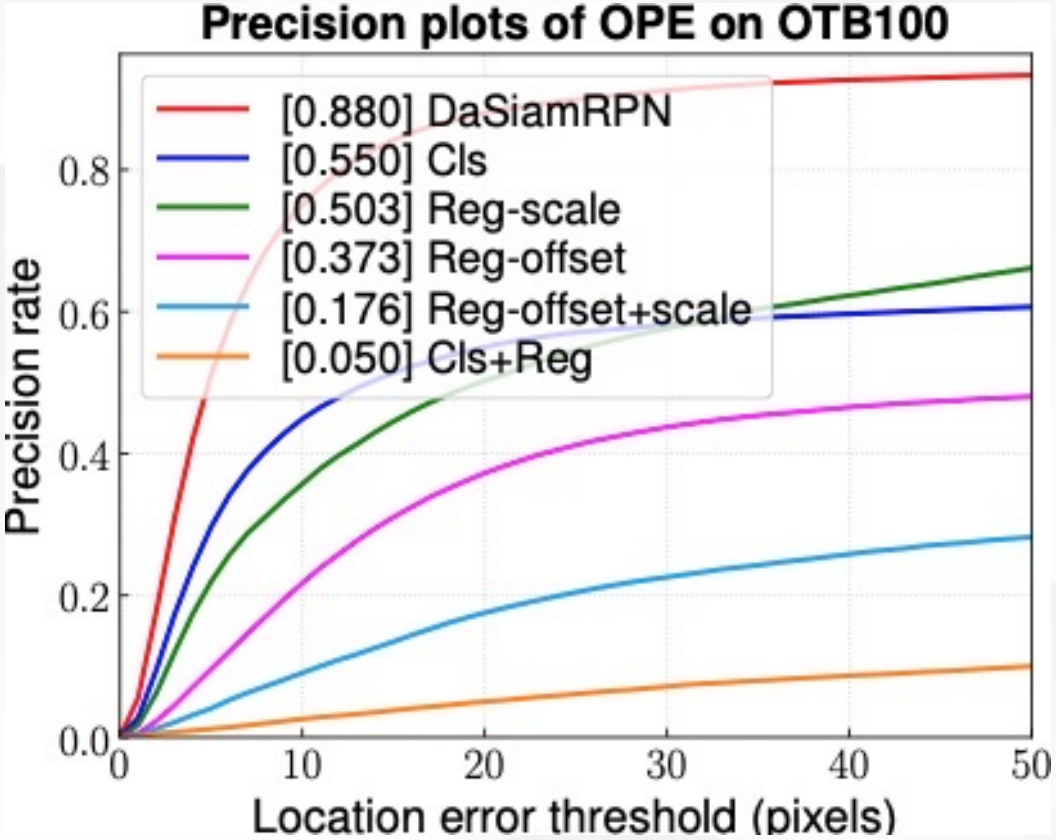
$$I_1^t = I_1^t - \gamma \cdot (I_1^{t-1} - I_M^{t-1})$$

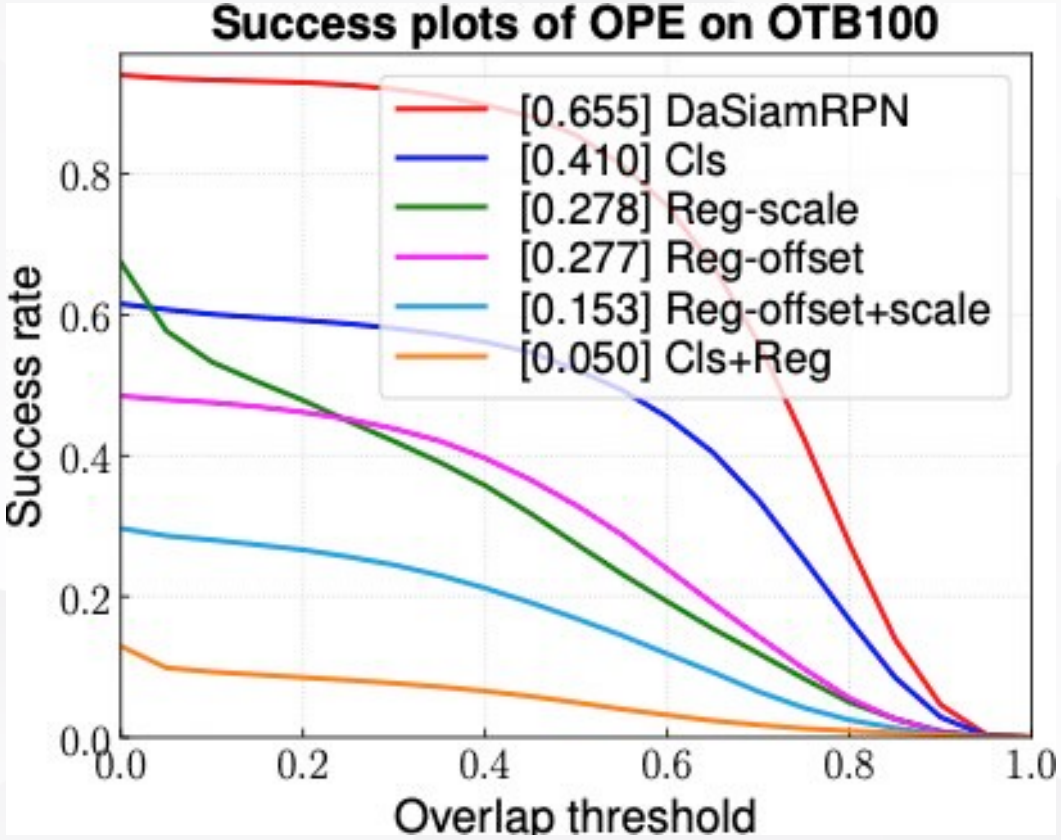
$$\begin{aligned} x_r^\star &= x_r + \delta_{\text{offset}} \\ y_r^\star &= y_r + \delta_{\text{offset}} \\ w_r^\star &= w_r * \delta_{\text{scale}} \\ h_r^\star &= h_r * \delta_{\text{scale}} \end{aligned}$$

$$\mathcal{L}_{adv}(I, N, \theta) = \sum_{n=1}^{N} \{[L_c(I_n, p_c, \theta) - L_c(I_n, p_c^\star, \theta)] + \lambda \cdot [L_r(I_n, p_r, \theta) - L_r(I_n, p_r^\star, \theta)]\}$$

$$I_{m+1} = I_m + \alpha \cdot sign(r_m)$$

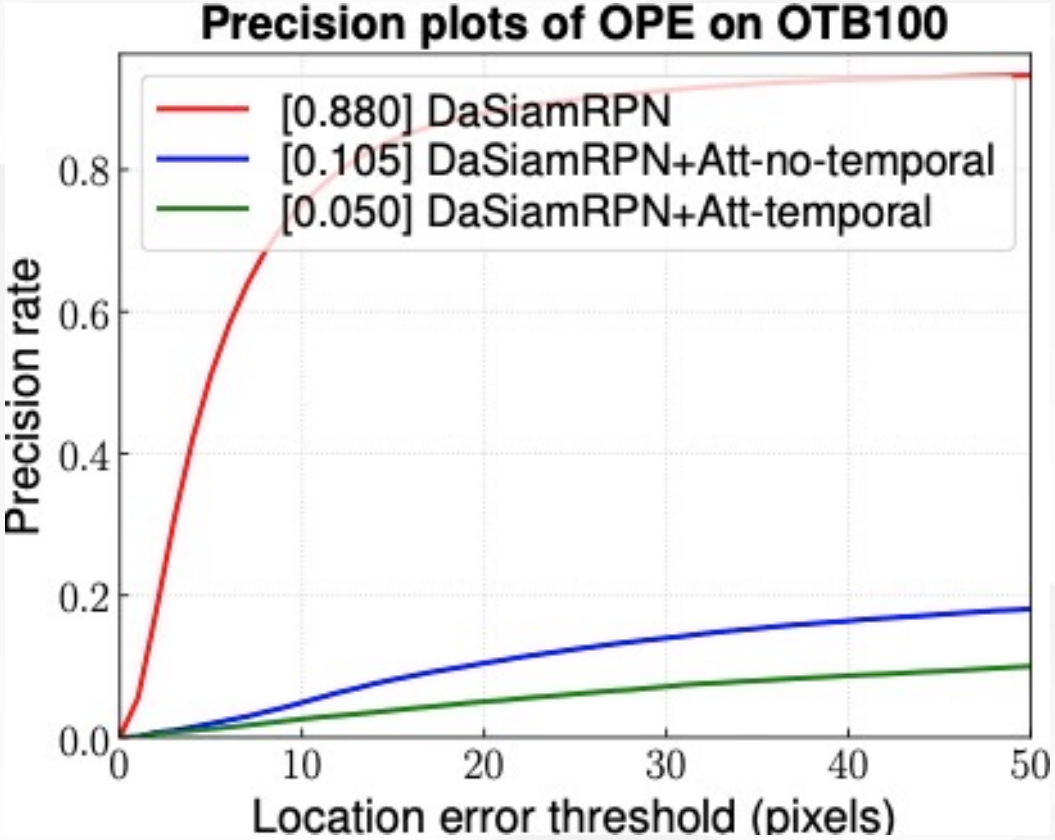
Experimental Results: Ablation Study

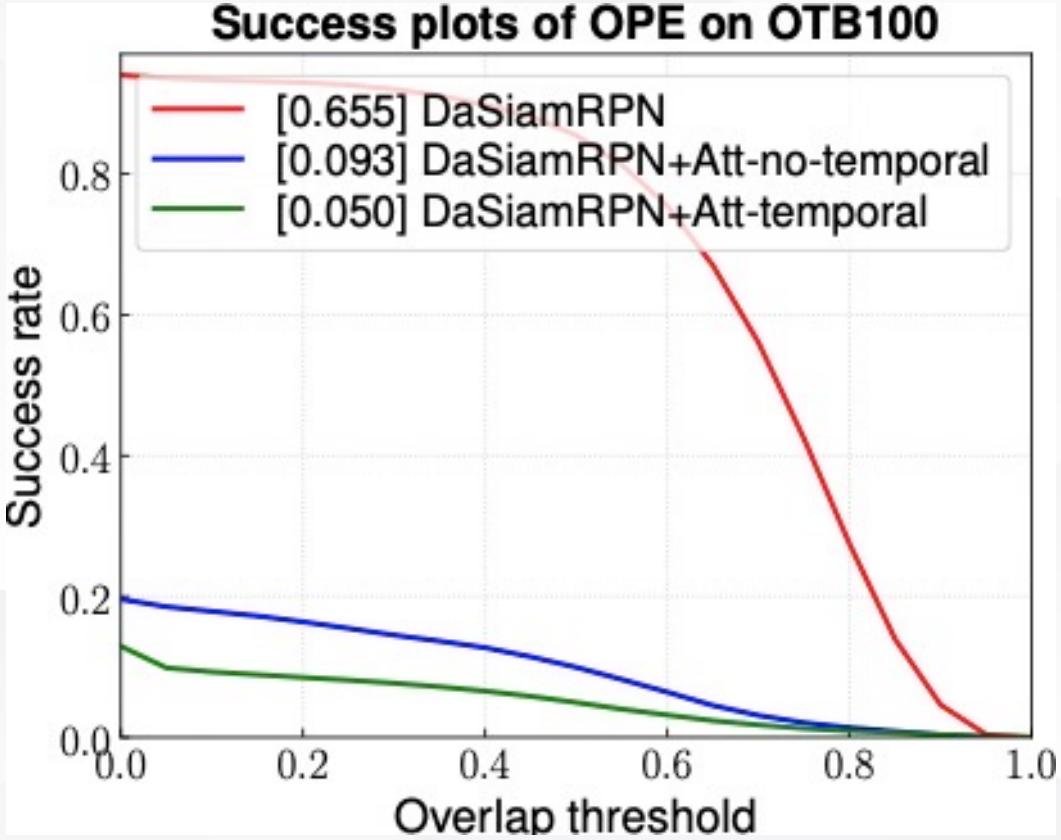




Ablation studies of DaSiamRPN on the OTB100 dataset. We denote Cls as the attack on the classification branch, Reg as the attack on the regression branch where there are offset and scale attacks.

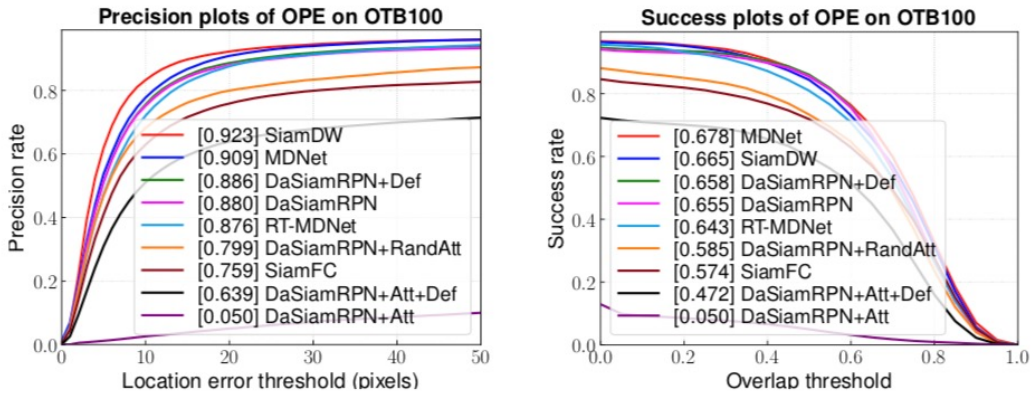
Experimental Results: Ablation Study



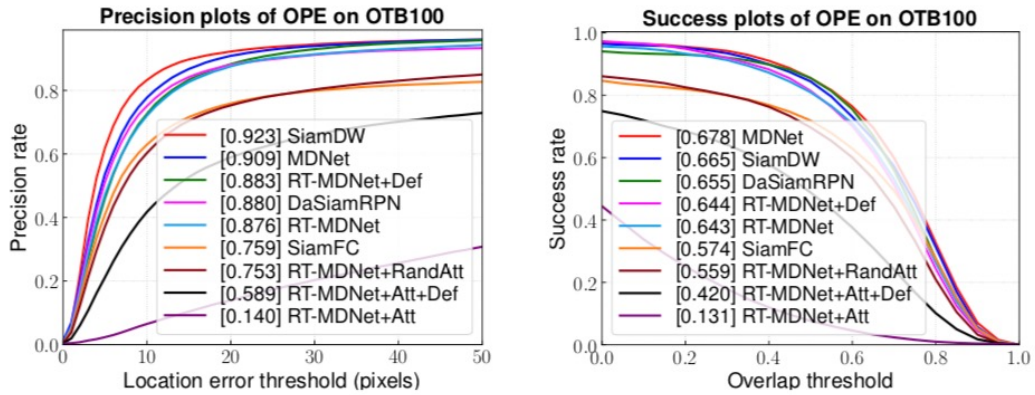


Ablation studies on temporal consistency of DaSiamRPN on the OTB-2015 dataset. Temporal denotes using temporal consistency in adversarial attack

Experiments

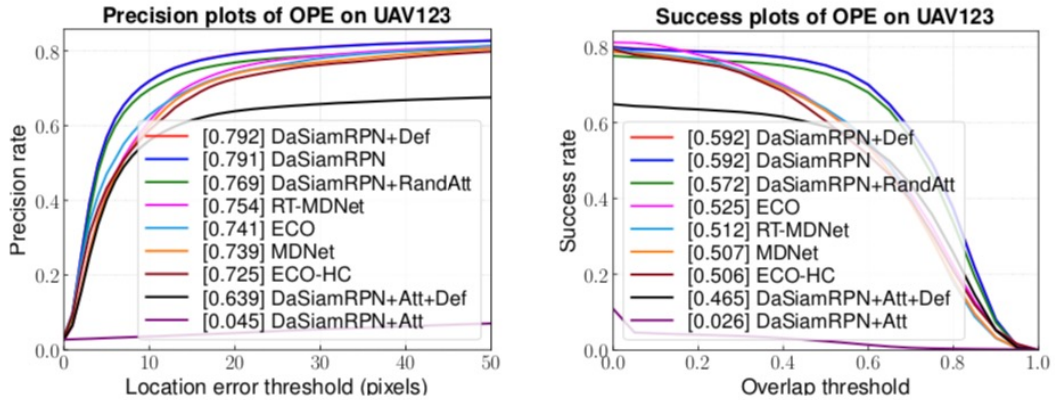


(a) DaSiamRPN

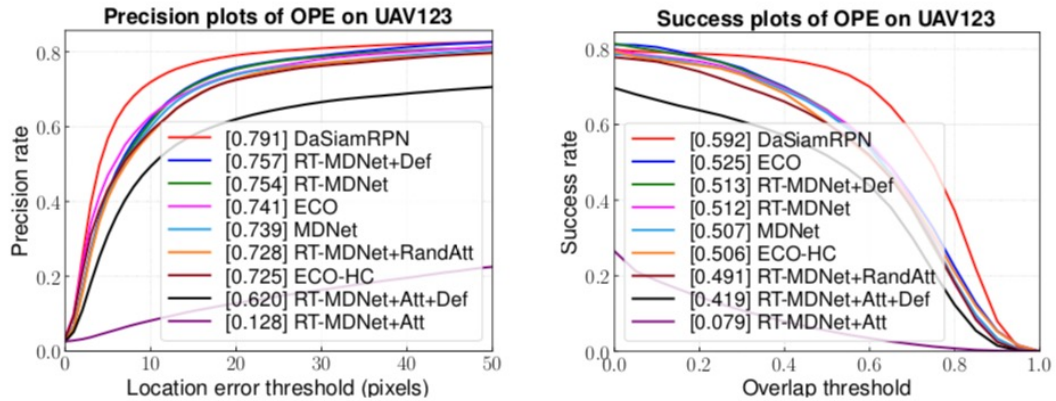


(b) RT-MDNet

Evaluations on the OTB100 dataset.



(a) DaSiamRPN



(b) RT-MDNet

Evaluations on the UAV123 dataset.

Experiments

	Accuracy ↑	Robustness ↓	Failures ↓	EAO ↑
DaSiamRPN	0.585	0.272	58	0.380
DaSiamRPN+RandAtt	0.571	0.529	113	0.223
DaSiamRPN+Att	0.536	1.447	309	0.097
DaSiamRPN+Att+Def	0.579	0.674	144	0.195
DaSiamRPN+Def	0.584	**0.253**	**54**	**0.384**

	Accuracy ↑	Robustness ↓	Failures ↓	EAO ↑
RT-MDNet	0.533	0.567	121	0.176
RT-MDNet+RandAtt	0.503	0.871	186	0.137
RT-MDNet+Att	0.475	1.611	344	0.076
RT-MDNet+Att+Def	0.515	1.021	218	0.110
RT-MDNet+Def	0.529	**0.538**	**115**	**0.179**

Evaluations on the VOT2018 dataset.

	Accuracy ↑	Robustness ↓	Failures ↓	EAO ↑
DaSiamRPN	0.625	0.224	48	0.439
DaSiamRPN+RandAtt	0.606	0.303	65	0.336
DaSiamRPN+Att	0.521	1.613	350	0.078
DaSiamRPN+Att+Def	0.581	0.722	155	0.211
DaSiamRPN+Def	0.622	**0.214**	**46**	0.418

	Accuracy ↑	Robustness ↓	Failures ↓	EAO ↑
RT-MDNet	0.567	0.196	42	0.370
RT-MDNet+RandAtt	0.550	0.452	97	0.235
RT-MDNet+Att	0.469	0.928	199	0.128
RT-MDNet+Att+Def	0.531	0.494	106	0.225
RT-MDNet+Def	0.540	**0.168**	**36**	**0.374**

Evaluations on the VOT2016 dataset.

Demos for adversarial attack and defense

人工智能研究院
Artificial Intelligence Institute

DaSiamRPN RT-MDNet Ground Truth

Videos from OTB100 dataset

Demos for adversarial defense on clean sequences

人工智能研究院
Artificial Intelligence Institute

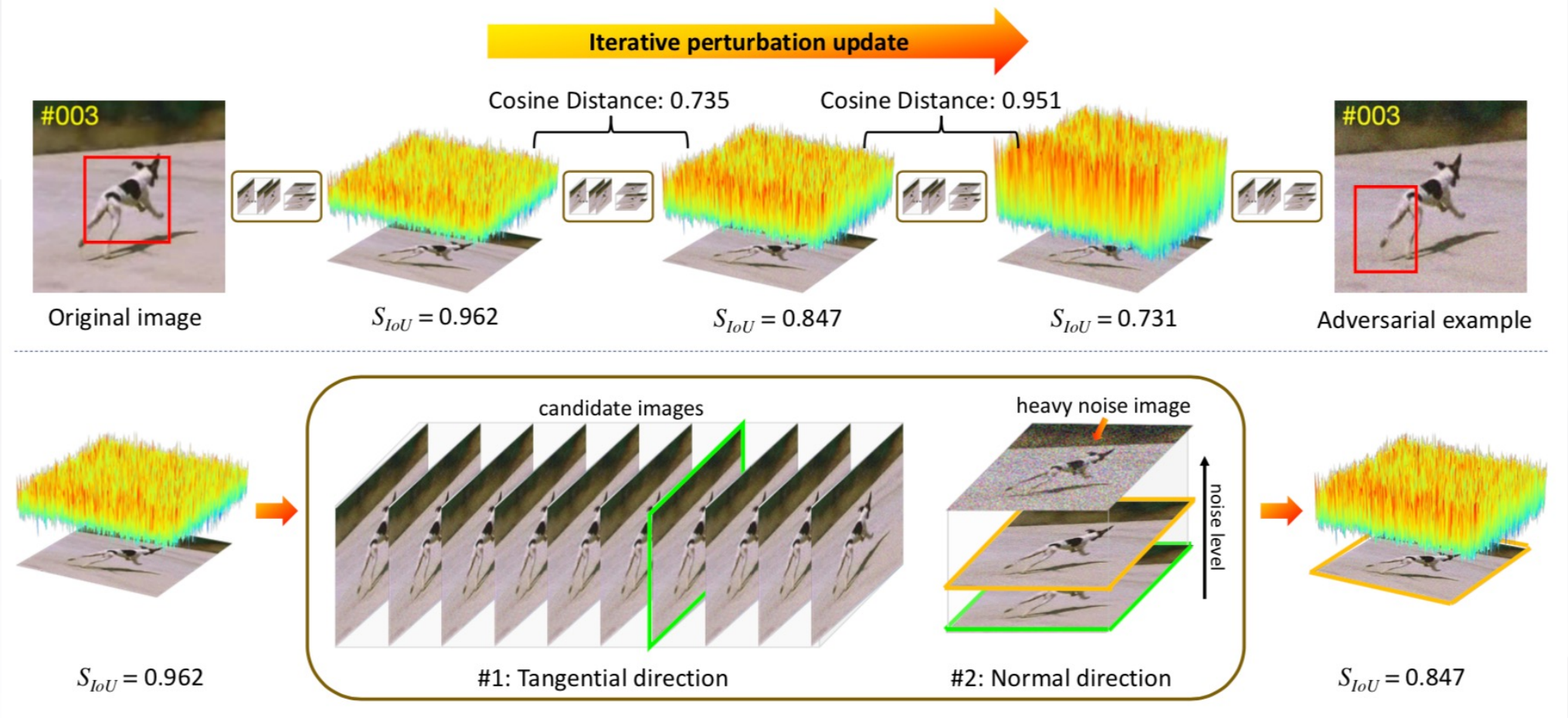
DaSiamRPN (baseline)　DaSiamRPN+Def　Ground Truth

Videos from OTB100 dataset

目　录

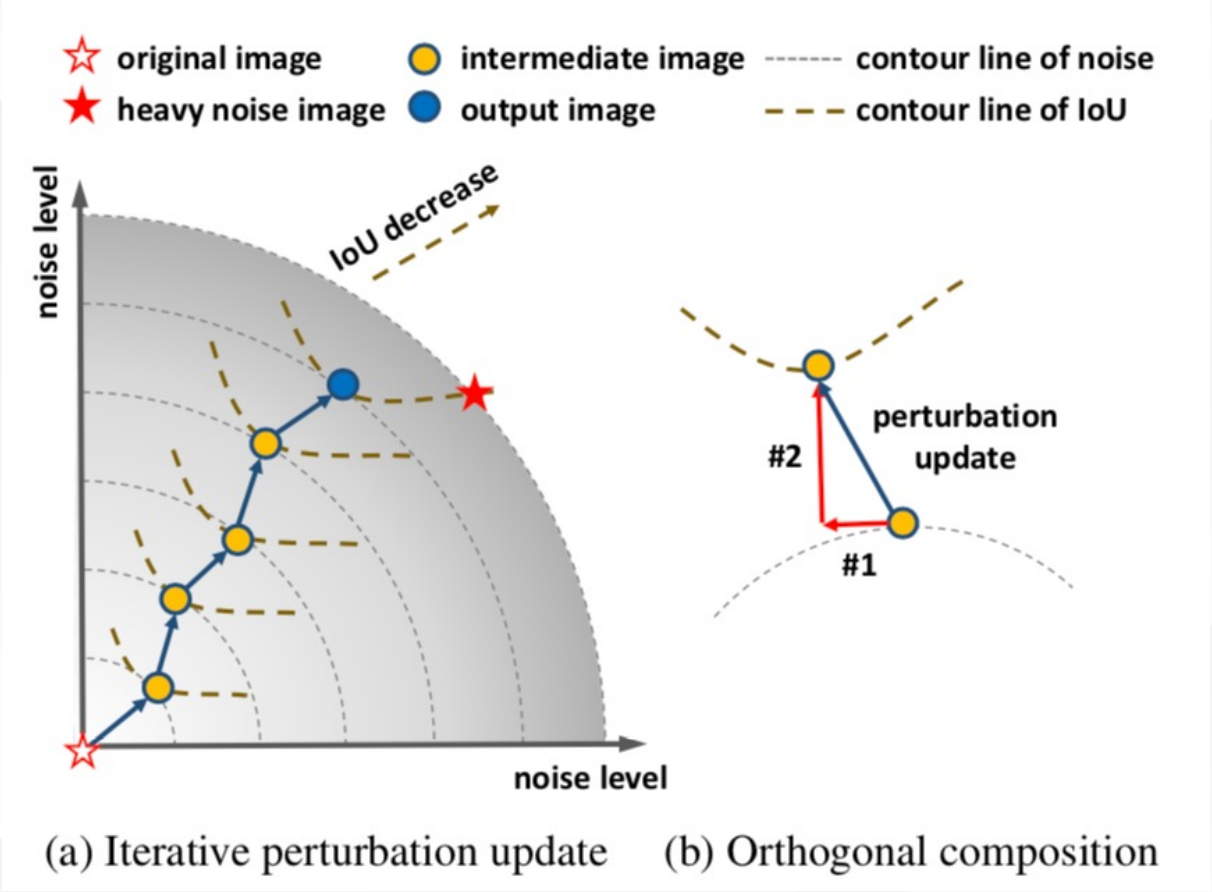
- 视觉跟踪算法白盒攻击（ECCV 2020）
- 目标跟踪算法黑盒攻击（CVPR 2021）

Method



- IoU attack aims to identify one specific noise perturbation leading to the lowest IoU score among the same amount of noise levels.

An intuitive view of IoU attack in the image space

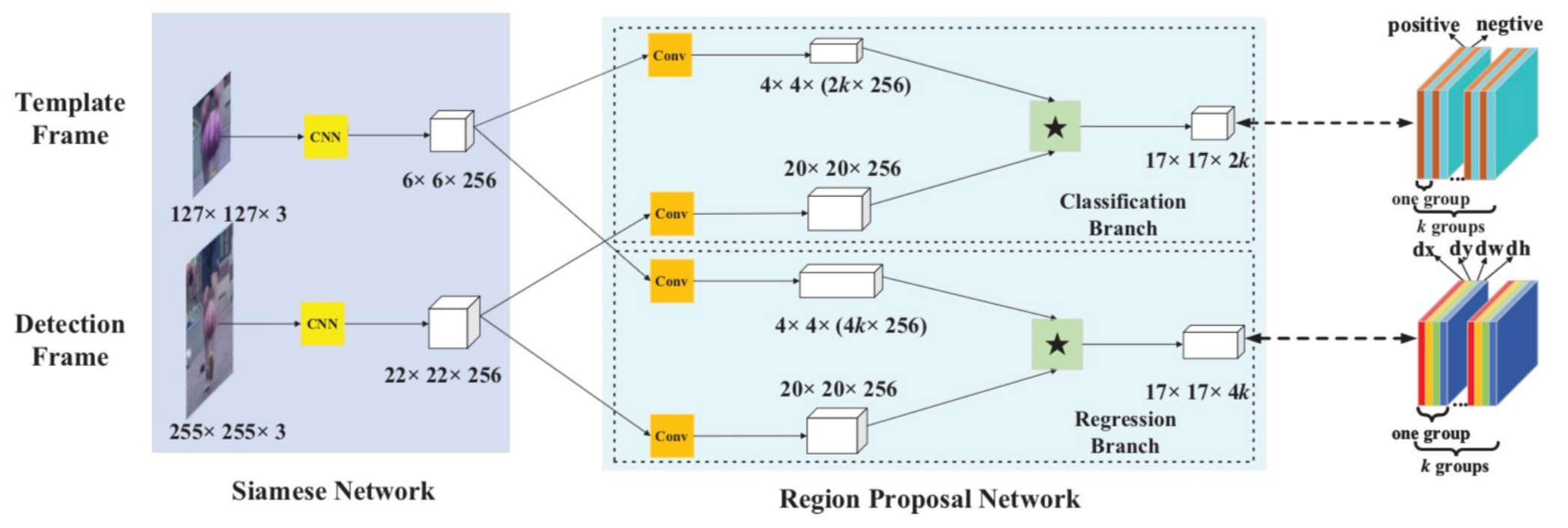


★ heavy noise image

☆ original image

CVPR 2021: IoU Attack: Towards Temporally Coherent Black-Box Adversarial Attack for Visual Object Tracking

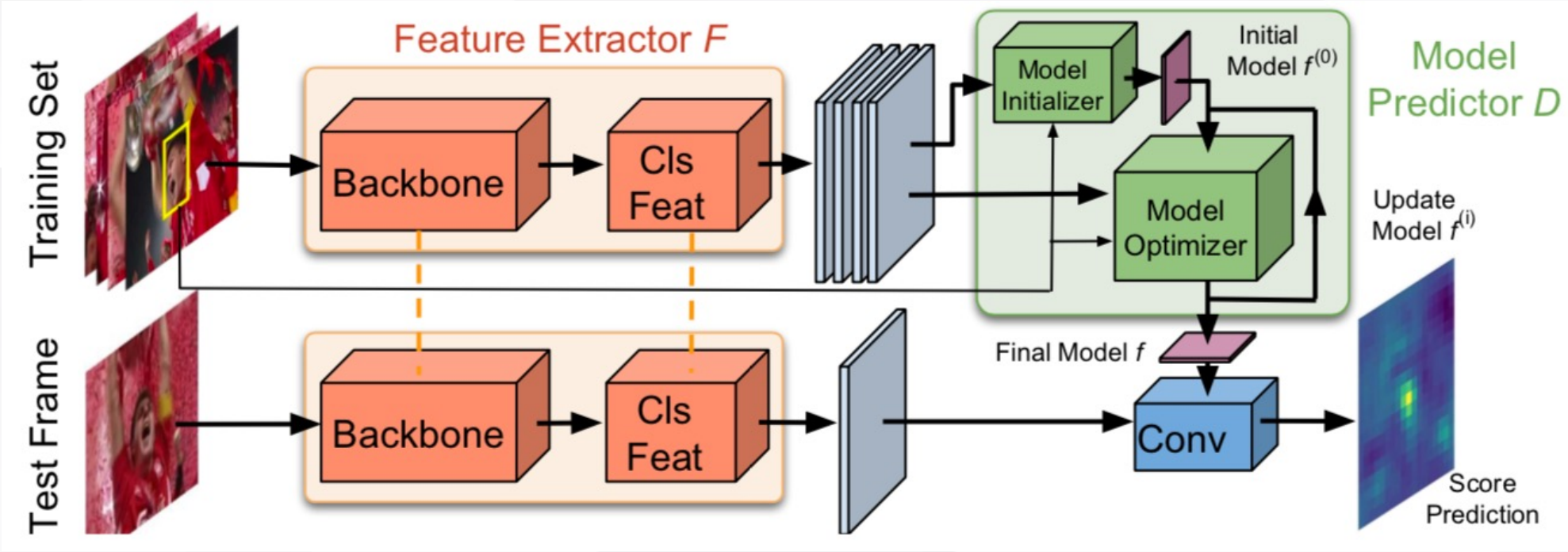
SiamRPN++ (Detection based, offline)



- SiamRPN++ is a end-to-end trained off-line tracker, consisting of Siamese subnetwork for feature extraction and region proposal subnetwork including the classification branch and regression branch.

[1] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In CVPR, 2019.

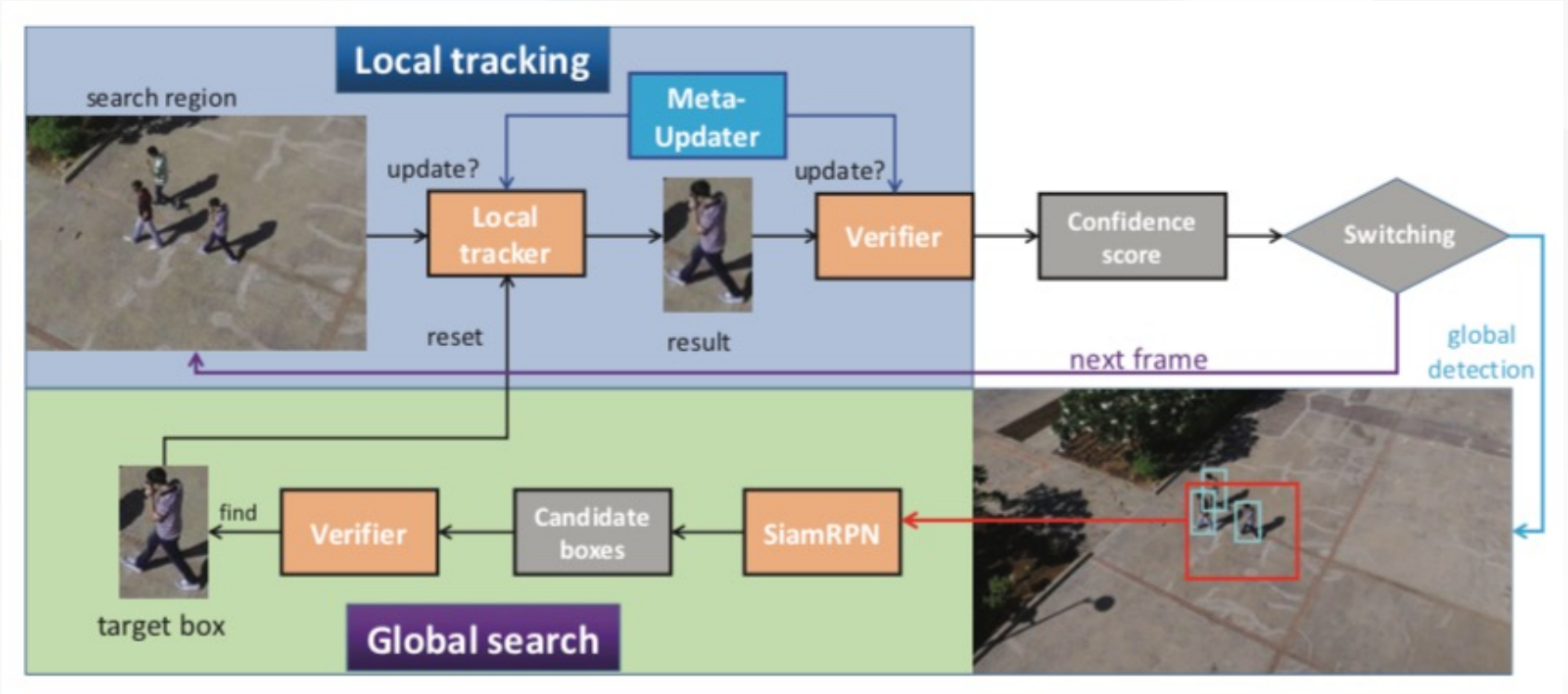
DiMP (Correlation filter based, online)



- DiMP exploits both target and background appearance information to locate the target by learning the discriminative target model during offline training and updating the optimization with only a few iterations.

[1] Bhat, G., Danelljan, M., Gool, L. V., Timofte, R. Learning discriminative model prediction for tracking. In: ICCV (2019)

LTMU (Long-term tracker)



- LTMU is specifically designed for long-term tracking and consists of a local tracker, an online verifier, a SiamRPN-based re-detector, and a meta-updater. The meta-updater learns to guide the tracker update properly.

[1] Dai, K., Zhang, Y., Wang, D., Li, J., Lu, H., Yang, X. High-performance long-term tracking with meta-updater. In: CVPR (2020).

Ablation Studies

Tracker	S_{temporal}	P_{t-1}	EAO ↑	
			VOT2018	VOT2016
SiamRPN++	No	Yes	0.149	0.189
	Yes	No	0.134	0.190
	Yes	Yes	**0.129**	**0.183**
DiMP	No	Yes	0.257	0.275
	Yes	No	0.261	0.295
	Yes	Yes	**0.248**	**0.256**
LTMU	No	Yes	0.147	0.184
	Yes	No	0.150	0.189
	Yes	Yes	**0.120**	**0.170**

- S_{temporal} **represents the temporal IoU score;**
- P_{t-1} **represents the learned perturbation from historical frames;**

Trackers	Accuracy ↑			Robustness ↓			Failures ↓			EAO ↑		
	Orig.	Rand.	Attack	Orig.	Rand.	Attack	Orig.	Rand.	Attack	Orig.	Rand.	Attack
SiamRPN++	0.596	0.591	**0.575**	0.472	0.727	**1.575**	94	145	**314**	0.287	0.220	**0.124**
DiMP	0.568	0.567	**0.474**	0.277	0.373	**0.641**	55	74	**127**	0.332	0.284	**0.195**
LTMU	0.625	0.623	**0.576**	0.913	1.073	**1.470**	182	214	**293**	0.201	0.175	**0.150**

Results on the VOT2019 dataset.

Trackers	Accuracy ↑			Robustness ↓			Failures ↓			EAO ↑		
	Orig.	Rand.	Attack	Orig.	Rand.	Attack	Orig.	Rand.	Attack	Orig.	Rand.	Attack
SiamRPN++	0.602	0.587	**0.568**	0.239	0.365	**1.171**	51	78	**250**	0.413	0.301	**0.129**
DiMP	0.574	0.560	**0.507**	0.145	0.202	**0.400**	31	43	**85**	0.427	0.363	**0.248**
LTMU	0.624	0.622	**0.590**	0.702	0.805	**1.320**	150	172	**282**	0.195	0.178	**0.120**

Results on the VOT2018 dataset.

Trackers	Accuracy ↑			Robustness ↓			Failures ↓			EAO ↑		
	Orig.	Rand.	Attack	Orig.	Rand.	Attack	Orig.	Rand.	Attack	Orig.	Rand.	Attack
SiamRPN++	0.643	0.632	**0.605**	0.200	0.340	**0.802**	43	73	**172**	0.461	0.331	**0.183**
DiMP	0.599	0.592	**0.536**	0.140	0.168	**0.374**	30	36	**80**	0.449	0.404	**0.256**
LTMU	0.661	0.646	**0.604**	0.522	0.592	**0.904**	112	127	**194**	0.236	0.233	**0.170**

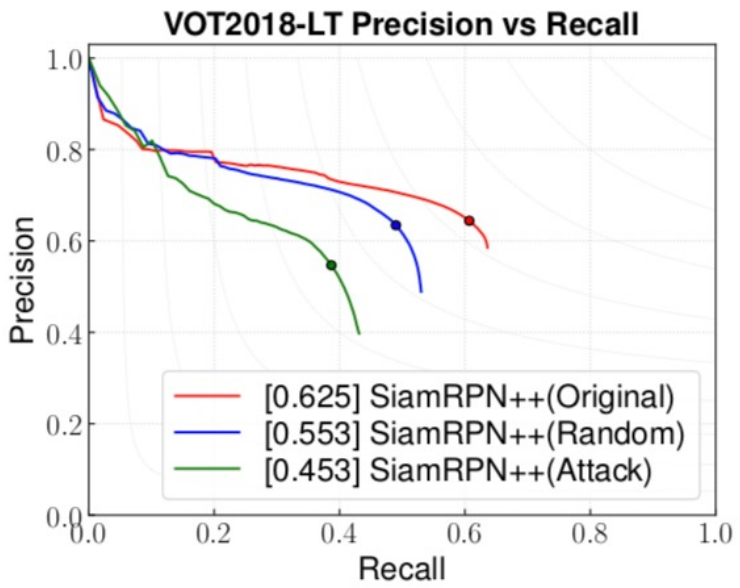
Results on the VOT2016 dataset.

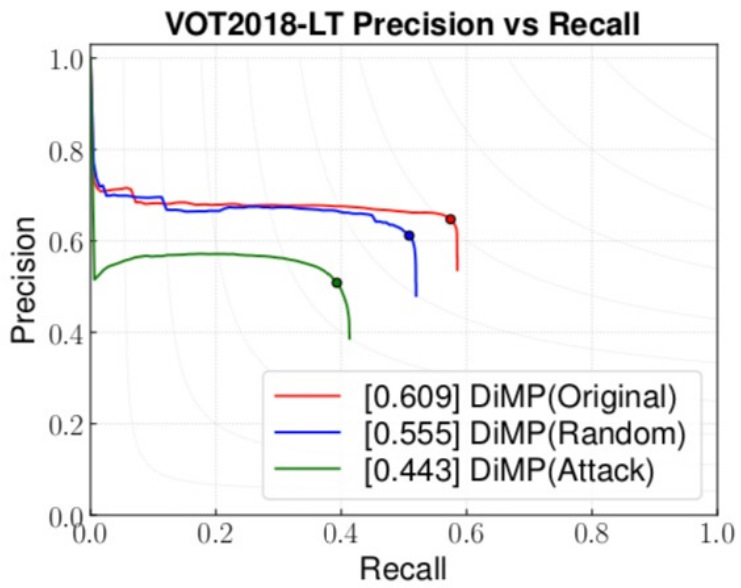
Trackers	Success ↑			Precision ↑		
	Orig.	Rand.	Attack	Orig.	Rand.	Attack
SiamRPN++	0.695	0.631	**0.499**	0.905	0.818	**0.644**
DiMP	0.671	0.659	**0.592**	0.869	0.860	**0.791**
LTMU	0.672	0.622	**0.517**	0.872	0.815	**0.712**

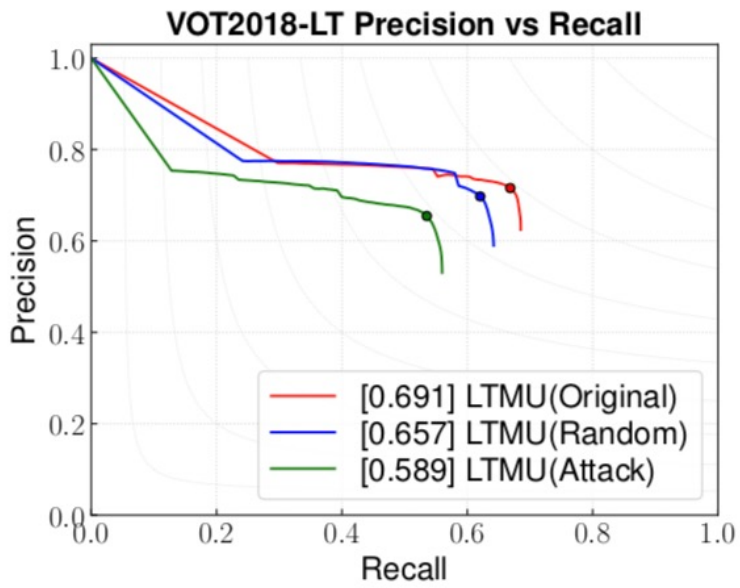
Results on the OTB100 dataset.

Trackers	Success ↑			Precision ↑		
	Orig.	Rand.	Attack	Orig.	Rand.	Attack
SiamRPN++	0.509	0.466	**0.394**	0.601	0.550	**0.446**
DiMP	0.614	0.591	**0.545**	0.729	0.710	**0.658**
LTMU	0.631	0.579	**0.462**	0.764	0.699	**0.559**

Results on the NFS30 dataset.







Results on the VOT2018-LT dataset.

SiamRPN++ DiMP LTMU

From the NFS30 dataset

Conclusions

- The performance of deep trackers degrades rapidly under attacks.
- White-box attacks are more aggressive than black-box attacks.
- Learning deep trackers with defense schemes can improve the tracking robustness.

Thanks!

ECCV 2020: https://github.com/VISION-SJTU/RTAA

CVPR 2021: https://github.com/VISION-SJTU/IoUattack