Adversarial Texture for Fooling Person Detectors in the Physical World

Supplementary Materials

Zhanhao Hu[1] Siyuan Huang[1] Xiaopei Zhu[2,1] Fuchun Sun[1] Bo Zhang[1] Xiaolin Hu[1,3,4*]
[1]Department of Computer Science and Technology, Institute for Artificial Intelligence,
State Key Laboratory of Intelligent Technology and Systems, BNRist, Tsinghua University, Beijing, China
[2]School of Integrated Circuits, Tsinghua University, Beijing, China
[3]IDG/McGovern Institute for Brain Research, Tsinghua University, Beijing, China
[4]Chinese Institute for Brain Research (CIBR), Beijing, China
{huzhanha17, zxp18}@mails.tsinghua.edu.cn
{siyuanhuang, fcsun, dcszb, xlhu}@mail.tsinghua.edu.cn

1. Proofs

1.1. Proof of Theorem 1

The KL divergence $\mathrm{KL}(q_\varphi(\tilde{\tau})||p_{adv}(\tilde{\tau}))$ can be divided into two terms:

$$\begin{aligned}\mathrm{KL}(q_\varphi(\tilde{\tau})||p_{adv}(\tilde{\tau})) &= \int_{\tilde{\tau}} q_\varphi(\tilde{\tau})\log\frac{q_\varphi(\tilde{\tau})}{p_{adv}(\tilde{\tau})}\,\mathrm{d}\tilde{\tau} \\ &= \int_{\tilde{\tau}} q_\varphi(\tilde{\tau})\log q_\varphi(\tilde{\tau})\,\mathrm{d}\tilde{\tau} - \int_{\tilde{\tau}} q_\varphi(\tilde{\tau})\log p_{adv}(\tilde{\tau})\,\mathrm{d}\tilde{\tau},\end{aligned} \tag{1}$$

where the first term is the negative entropy of q_φ, i.e., $-\mathrm{H}_\varphi(\tilde{\tau})$. We introduce mutual information (MI) to help compute the entropy:

$$\mathrm{I}_\varphi(\tilde{\tau}, z) = \int_{\tilde{\tau},z} p(\tilde{\tau}, z)\log\frac{p(\tilde{\tau}, z)}{q_\varphi(\tilde{\tau})p_z(z)}\,\mathrm{d}\tilde{\tau}\,\mathrm{d}z, \tag{2}$$

where $p(\tilde{\tau}, z)$ is the joint distribution of $\tilde{\tau} = G_\varphi(z)$ and z. Since $p(\tilde{\tau}, z) = p(\tilde{\tau}|z)p_z(z)$ and $q_\varphi(\tilde{\tau})$ is the marginal distribution $q_\varphi(\tilde{\tau}) = \int_z p(\tilde{\tau}, z)\,\mathrm{d}z$, we have

$$\begin{aligned}\mathrm{I}_\varphi(\tilde{\tau}, z) &= \int_{\tilde{\tau},z} p(\tilde{\tau}, z)\log\frac{p(\tilde{\tau}, z)}{p_z(z)}\,\mathrm{d}\tilde{\tau}\,\mathrm{d}z - \int_{\tilde{\tau},z} p(\tilde{\tau}, z)\log q_\varphi(\tilde{\tau})\,\mathrm{d}\tilde{\tau}\,\mathrm{d}z \\ &= \int_{\tilde{\tau},z} p(\tilde{\tau}|z)p_z(z)\log p(\tilde{\tau}|z)\,\mathrm{d}\tilde{\tau}\,\mathrm{d}z - \int_{\tilde{\tau}} \log q_\varphi(\tilde{\tau})\,\mathrm{d}\tilde{\tau}\int_z p(\tilde{\tau}, z)\,\mathrm{d}z \\ &= \int_z p_z(z)\int_{\tilde{\tau}} p(\tilde{\tau}|z)\log p(\tilde{\tau}|z)\,\mathrm{d}\tilde{\tau}\,\mathrm{d}z - \int_{\tilde{\tau}} q_\varphi(\tilde{\tau})\log q_\varphi(\tilde{\tau})\,\mathrm{d}\tilde{\tau} \\ &= -\,\mathrm{H}_\varphi(\tilde{\tau}|z) + \mathrm{H}_\varphi(\tilde{\tau}),\end{aligned} \tag{3}$$

*Corresponding author.

where $\mathrm{H}_\varphi(\tilde{\tau}|z)$ is called conditional entropy. Therefore, the first term of Eq. (1) can be replaced by $-\mathrm{I}_\varphi(\tilde{\tau}, z) - \mathrm{H}_\varphi(\tilde{\tau}|z)$. Since $\tilde{\tau} \sim q_\varphi$ is determined by z, i.e., $p(\tilde{\tau}|z) = \delta(\tilde{\tau} - G_\varphi(z))$, we have

$$\begin{aligned}
\mathrm{H}_\varphi(\tilde{\tau}|z) &= -\int_z p_z(z) \int_{\tilde{\tau}} p(\tilde{\tau}|z) \log p(\tilde{\tau}|z) \,\mathrm{d}\tilde{\tau}\,\mathrm{d}z \\
&= -\int_z p_z(z) \int_{\tilde{\tau}} \delta(\tilde{\tau} - G_\varphi(z)) \log \delta(\tilde{\tau} - G_\varphi(z)) \,\mathrm{d}\tilde{\tau}\,\mathrm{d}z \\
&= -\int_z p_z(z)\,\mathrm{d}z \int_{\tilde{\tau}'} \delta(\tilde{\tau}') \log \delta(\tilde{\tau}')\,\mathrm{d}\tilde{\tau}' \qquad (4) \\
&= -\int_{\tilde{\tau}'} \delta(\tilde{\tau}') \log \delta(\tilde{\tau}')\,\mathrm{d}\tilde{\tau}', \qquad (5)
\end{aligned}$$

which indicates that $\mathrm{H}_\varphi(\tilde{\tau}|z)$ is a constant[1]. Therefore, we ignore this term in Eq. (3). Moreover, for the second term of Eq. (1), since $p_{adv}(\tilde{\tau}) = \frac{e^{-U(\tilde{\tau})}}{Z_U}$, we have

$$\begin{aligned}
-\int_{\tilde{\tau}} q_\varphi(\tilde{\tau}) \log p_{adv}(\tilde{\tau})\,\mathrm{d}\tilde{\tau} &= -\int_{\tilde{\tau}} q_\varphi(\tilde{\tau}) \log \frac{e^{-U(\tilde{\tau})}}{Z_U}\,\mathrm{d}\tilde{\tau} \\
&= \int_{\tilde{\tau}} q_\varphi(\tilde{\tau}) U(\tilde{\tau})\,\mathrm{d}\tilde{\tau} + \int_{\tilde{\tau}} q_\varphi(\tilde{\tau}) \log Z_U\,\mathrm{d}\tilde{\tau} \\
&= \mathbb{E}_{\tilde{\tau}\sim q_\varphi(\tilde{\tau})}[U(\tilde{\tau})] + \log Z_U, \qquad (6)
\end{aligned}$$

where the partition function $Z_U = \int_{\tilde{\tau}} e^{-U(\tilde{\tau})} \mathrm{d}\tilde{\tau}$ is a constant.

Therefore, minimizing Eq. (1) is equivalent to

$$\min_\varphi -\mathrm{I}_\varphi(\tilde{\tau}, z) + \mathbb{E}_{\tilde{\tau}\sim q_\varphi(\tilde{\tau})}[U(\tilde{\tau})]. \qquad (7)$$

In other words, we need to simultaneously maximize $\mathrm{I}_\varphi(\tilde{\tau}, z)$ and minimize $\mathbb{E}_{\tilde{\tau}\sim q_\varphi(\tilde{\tau})}[U(\tilde{\tau})]$. According to Deep InfoMax (DIM) [4], maximizing $\mathrm{I}_\varphi(\tilde{\tau}, z)$ is equivalent to maximizing a Jensen-Shannon mutual information (MI) estimator,

$$\mathcal{I}^{\mathrm{JSD}}_{\varphi,\omega}(\tilde{\tau}, z) = \mathbb{E}_{(\tilde{\tau},z)\sim q_\varphi^{\tilde{\tau},z}(\tilde{\tau},z)}[-\mathrm{sp}(-T_\omega(\tilde{\tau}, z))] - \mathbb{E}_{\tilde{\tau}\sim q_\varphi(\tilde{\tau}), z'\sim p_z(z')}[\mathrm{sp}(T_\omega(\tilde{\tau}, z'))], \qquad (8)$$

where $q_\varphi^{\tilde{\tau},z}$ denotes the joint distribution of $\tilde{\tau}$ and z, and $\mathrm{sp}(t) = \log(1 + e^t)$ is the softplus function. T_ω is a scalar function modeled by a neural network whose parameter ω must be optimized together with the parameter φ. Therefore, we replace $\mathrm{I}_\varphi(\tilde{\tau}, z)$ by $\mathcal{I}^{\mathrm{JSD}}_{\varphi,\omega}(\tilde{\tau}, z)$ and optimize φ and ω simultaneously.

Given the above, minimizing $\mathrm{KL}(q_\varphi(\tilde{\tau})||p_{adv}(\tilde{\tau}))$ is equivalent to

$$\min_{\varphi,\omega} -I^{\mathrm{JSD}}_{\varphi,\omega}(\tilde{\tau}, z) + \mathbb{E}_{\tilde{\tau}\sim q_\varphi(\tilde{\tau})}[U(\tilde{\tau})]. \qquad (9)$$

1.2. Proof of Theorem 2

Since G_1 is equivalent to G_2, τ_1 has the same dimension as τ_2.We denote the dimension by K. Let τ_1^k be the k-th element of τ_1, and τ_2^k be the k-th element of τ_2. Since $\mathcal{Z}_1$ is identical to $\mathcal{Z}_2$, i.e. the probability density function (PDF) $p_{\mathcal{Z}_1}(z) = p_{\mathcal{Z}_2}(z)$, we have

$$\begin{aligned}
&\Pr(\tau_1^k < h_k, k = 1, 2, ..., K) \\
&= \int_{G_1(z)_k < h_k, k=1,2,...,K} p_{\mathcal{Z}_1}(z)\mathrm{d}z \\
&= \int_{G_2(z)_k < h_k, k=1,2,...,K} p_{\mathcal{Z}_2}(z)\mathrm{d}z \\
&= \Pr(\tau_2^k < h_k, k = 1, 2, ..., K), \qquad (10)
\end{aligned}$$

where $\{h_k\}_{k=1,2,...,K}$ is a list of arbitrary real numbers. Therefore, the cumulative distribution function (CDF) of $\mathcal{T}_1$ is equal to the CDF of $\mathcal{T}_2$, which proves that $\mathcal{T}_1$ is identical to $\mathcal{T}_2$.

[1] In fact, it is zero for discrete distribution and is infinity for continuous distribution.

1.3. Proof of Corollary 2.1

Assuming that the FCN has L layers, we define $\mathrm{Conv}^{(l)}$, $\mathrm{Kernel}^{(l)}$ and $\mathrm{Act}^{(l)}$ as the convolutional function, the convolutional kernel and the element-wise activation function at the lth layer, respectively. Let the spatial size of $\mathrm{Kernel}^{(l)}$ be $a^{(l)}$ and $b^{(l)}$. We denote the value before the activation function at the lth layer by $o^{(l)}$ and denote the feature map by $v^{(l)}$. We further define $v^{(0)}$ as the input z and define $v^{(L)}$ as the output τ. Therefore, for $l \in \{1, 2, ..., L\}$, we have

$$o^{(l)} = \mathrm{Conv}^{(l)}(v^{(l-1)}) = v^{(l-1)} * \mathrm{Kernel}^{(l)}, \tag{11}$$

$$v^{(l)} = \mathrm{Act}^{(l)}(o^{(l)}), \tag{12}$$

where the operation $*$ stands for convolution. We denote $v^{(l)}_{i,j,w,h}$ and $o^{(l)}_{i,j,w,h}$ as a rectangular area with size $w \times h$ whose center is at the location i, j in $v^{(l)}$ and $o^{(l)}$ respectively. Ignoring the boundary conditions, for all l, i, j, i', j', w, h, by the nature of the convolutional operation, we have

$$o^{(l)}_{i,j,w^{(l)},h^{(l)}} = v^{(l-1)}_{i,j,w^{(l-1)},h^{(l-1)}} * \mathrm{Kernel}^{(l)}, \tag{13}$$

$$v^{(l)}_{i,j,w^{(l)},h^{(l)}} = \mathrm{Act}^{(l)}(o^{(l)}_{i,j,w^{(l)},h^{(l)}}), \tag{14}$$

and

$$o^{(l)}_{i',j',w^{(l)},h^{(l)}} = v^{(l-1)}_{i',j',w^{(l-1)},h^{(l-1)}} * \mathrm{Kernel}^{(l)}, \tag{15}$$

$$v^{(l)}_{i',j',w^{(l)},h^{(l)}} = \mathrm{Act}^{(l)}(o^{(l)}_{i',j',w^{(l)},h^{(l)}}), \tag{16}$$

where $w^{(l-1)} = w^{(l)} + a^{(l)} - 1$, $h^{(l-1)} = h^{(l)} + b^{(l)} - 1$, $w^{(L)} = w$, and $h^{(L)} = h$. Therefore, we can define a function $G^{(l)}_{i,j,w,h}$ as $G^{(l)}_{i,j,w,h}(v^{(0)}_{i,j,w^{(0)},h^{(0)}}) = v^{(l)}_{i,j,w^{(l)},h^{(l)}}$.

When $l = 0$, $G^{(l)}_{i,j,w,h}$ is obviously equivalent to $G^{(l)}_{i,j,w,h}$, since they are both identical functions. Moreover, the distribution of $v^{(l)}_{i,j,w^{(l)},h^{(l)}}$ is also identical to $v^{(l)}_{i',j',w^{(l)},h^{(l)}}$, since each element of $v^{(0)}$ is independent and identically distributed.

For $l > 0$, we assume that $G^{(l-1)}_{i,j,w,h}$ is equivalent to $G^{(l-1)}_{i,j,w,h}$, and the distribution of $v^{(l-1)}_{i,j,w^{(l-1)},h^{(l-1)}}$ is identical to $v^{(l-1)}_{i',j',w^{(l-1)},h^{(l-1)}}$ for all i, j, i', j', w, h. According to Eqs. (13) to (16), for all $v^{(0)}_{i,j,w^{(0)},h^{(0)}} = v^{(0)}_{i',j',w^{(0)},h^{(0)}}$,

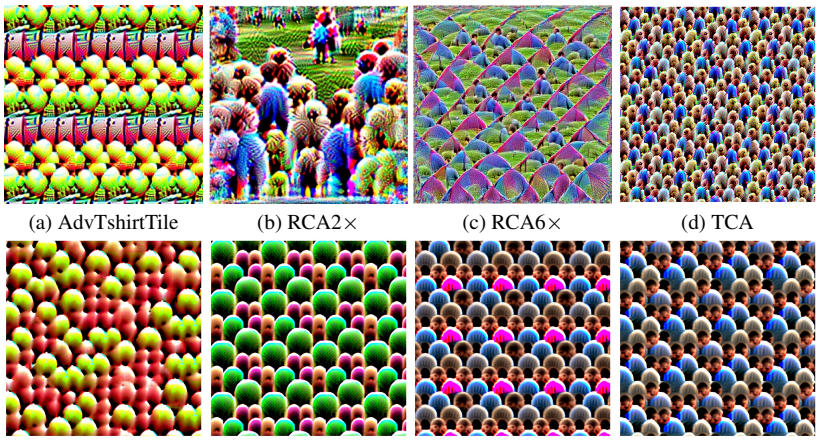
$$\begin{aligned}
G^{(l)}_{i,j,w,h}(v^{(0)}_{i,j,w^{(0)},h^{(0)}}) &= \mathrm{Act}^{(l)}(v^{(l-1)}_{i,j,w^{(l-1)},h^{(l-1)}} * \mathrm{Kernel}^{(l)}) \\
&= \mathrm{Act}^{(l)}(G^{(l-1)}_{i,j,w,h}(v^{(0)}_{i,j,w^{(0)},h^{(0)}})) * \mathrm{Kernel}^{(l)}) \\
&= \mathrm{Act}^{(l)}(G^{(l-1)}_{i',j',w,h}(v^{(0)}_{i',j',w^{(0)},h^{(0)}})) * \mathrm{Kernel}^{(l)}) \\
&= \mathrm{Act}^{(l)}(v^{(l-1)}_{i',j',w^{(l-1)},h^{(l-1)}} * \mathrm{Kernel}^{(l)}) \\
&= G^{(l)}_{i',j',w,h}(v^{(0)}_{i',j',w^{(0)},h^{(0)}}).
\end{aligned}$$

Therefore, $G^{(l)}_{i,j,w,h}$ is equivalent to $G^{(l)}_{i',j',w,h}$. Moreover, since the convolutions in Eqs. (13) and (15) are equivalent, the distribution of $o^{(l)}_{i,j,w^{(l)},h^{(l)}}$ is identical to that of $o^{(l)}_{i',j',w^{(l)},h^{(l)}}$ according to Theorem 2. Furthermore, Since the active function $\mathrm{Act}^{(l)}$ is element-wise, i.e., it is equivalent for i, j and i', j', the distribution of $v^{(l)}_{i,j,w^{(l)},h^{(l)}}$ is also identical to that of $v^{(l)}_{i',j',w^{(l)},h^{(l)}}$ according to Theorem 2.

By using mathematical induction, we conclude that $G^{(l)}_{i,j,w,h}$ is equivalent to $G^{(l)}_{i',j',w,h}$, and the distribution of $v^{(l)}_{i,j,w^{(l)},h^{(l)}}$ is identical to that of $v^{(l)}_{i',j',w^{(l)},h^{(l)}}$ for all $l \in [1, 2, ..., L]$. Since $v^{(L)} = \tau$, $w^{(l)} = w$, and $h^{(l)} = h$, we derive Corollary 2.1. Note that every convolutional layer of FCN needs to be zero-padded to avoid the boundary problem.

2. Adversarial Textures and Adversarial Clothes

Figs. S1 and S2 present additional adversarial textures and adversarial clothes, respectively, that are not presented in the main paper (Figs. 6 and 8) due to the page limit. Unless otherwise specified, all results about physical attacks presented in both the main paper and the *Supplementary Materials* were obtained by adversarial T-shirts.



(a) AdvTshirtTile (b) RCA2× (c) RCA6× (d) TCA

(e) EGA (f) YOLOv3 [8] (g) FasterRCNN [9] (h) MaskRCNN [3]

Figure S1. Visualization of different adversarial textures, extending Fig. 6 in the main paper. (a) The texture formed by tiling an adversarial patches [10] repeatedly. (b-e) The textures produced by different methods to attack YOLOv2 [7]. (f-h) The textures produced by TC-EGA to attack different detectors respectively.

(a) Random (b) AdvPatch (c) AdvTshirt

(d)YOLOv2 skirt (e) YOLOv3 (f) FasterRCNN (g) MaskRCNN

Figure S2. Real-world adversarial clothes produced by different methods, extending Fig. 8 in the main paper.

Target detector	YOLOv3	FasterRCNN	MaskRCNN
AP	0.511	0.419	0.492

Table S1. The APs of different detectors attacked by TC-EGA on Inria test set.

3. Results of attacking different detectors in the digital world

Tab. S1 presents the APs of YOLOv3, FasterRCNN and MaskRCNN on Inria test set. Note that the AP of each detector on the original test images is 1.0. Though these AdvTextures were not effective as that of YOLOv2 whose AP was 0.362 (See Tab. 1 in the main paper), they had lowered the AP of clean images by half.

4. Comparison between Indoor and Outdoor Conditions

We compared the attack effectiveness of different adversarial T-shirts in the indoor and outdoor scenes. We used the videos described in Sec.4.2 in the main paper. We extracted 32 frames from each video with viewing angles varying from $0°$ to $3°$. Therefore we collected $3 \times 32 = 96$ frames for each scene and each detector. The results are presented in Tab. S2. The indoor mASR was comparable to the outdoor mASR for each piece of adversarial clothing. It indicates that the adversarial clothes are effective in different scenes.

scene \ Target	YOLOv2	YOLOv3	FasterRCNN	MaskRCNN
Indoor	0.771	0.764	0.912	0.832
Outdoor	0.714	0.638	0.948	0.878

Table S2. The mASRs of the attacks at different distances between persons and camera.

5. Effectiveness of the Attack with Respect to the Distance to the Camera

We recorded additional videos for each person wearing YOLOv2 T-shirt in both indoor and outdoor scenes. The persons still turned a circle slowly in front of the camera to collect frames at different viewing angles. We varied the distance between the camera and the persons to be 1.6 m, 2.0 m, 2.6 m, 3.4 m, 4.4 m, 5.6 m, and 7.0 m. For each distance, we collected 3(persons) $\times$ 2(scenes) $\times$ 32(frames per video) $=$ 192 frames in total. Fig. S3 presents the mASRs of YOLOv2 T-shirt at various distances. The mASR was the highest when the persons was close to the camera (1.6 m, mASR 0.791). It decreased to 0.257 when the distance was 7.0 m.

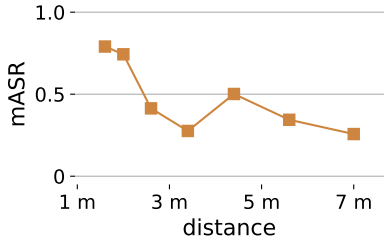


Figure S3. The mASRs of the attacks at different distances between persons and camera.

6. Attacking YOLOv3

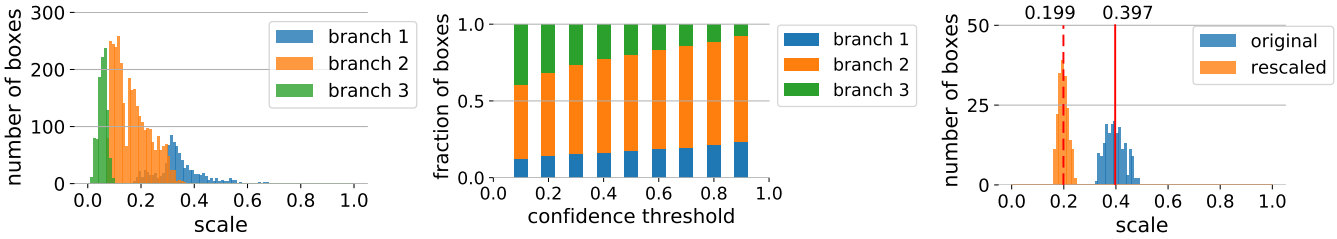
In this section we provide the reasons of scaling the size of the input by 50% before sending to YOLOv3 (see Tab. 4 in the main paper). YOLOv3 has three branches to predict boxes in different scales. These branches are based on feature maps of an backbone network in different layers, and use additional blocks before predicting boxes. Therefore, These branches are relatively independent when being adversarially attacked. Since the number of the boxes predicted by different branches can be quite different, the attack might be biased to one particular branch. Fig. S4a presents the histogram of the predicted boxes of each branch on the Inria training dataset, with a confidence threshold 0.5. The first branch predicted large scale boxes, and the third predicted small scale boxes. Fig. S4b presents the fraction of the predicted boxes with respect to different confidence thresholds. From the figure, the second branch predicted most of the boxes (62.8% when the confidence threshold is 0.5), indicating that the produced adversarial pattern may be biased towards attacking the second branch. However, in our recorded videos, the scale of the persons were outside the range of the second branch's predicted boxes (compare Figs. S4a and S4c). Therefore, we scaled the size of the input by 50% before sending the frames to YOLOv3.

7. Transfer Study in the Physical World

We performed transfer-based attacks on several detectors by the adversarial clothes that are produced to attack particular detectors. Tab. S3 presents the mASR of the transfer-based attacks. Every number in the table was obtained over 192 frames as described in Section 4.2 in the main paper. The adversarial clothes of YOLOv2 and YOLOv3 remained effective when they were used to attack YOLOv3 and YOLOv2, respectively. However, these clothes got low mASRs when attacking other models except RetinaNet. The adversarial clothes of Faster RCNN and MaskRCNN remained effective when they were used to attack other models, though sometimes (e.g., attacking YOLOv3) not as effective as attacking themselves. A possible solution is to use the model ensemble technique [2, 6], which is left as future research.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 6



(a) (b) (c)

Figure S4. (a) The distribution of the boxes' scales predicted by different branches of YOLOv3. For each box with normalized size $w \times h$, we define the scale by $\sqrt{w * h}$. (b) The fractions of the boxes predicted by different branches with respect to various confidence thresholds. (c) The distribution of the scales of the boxes on the original and rescaled video frames. The red solid line denotes the average scale on the original video frames, and the red dashed line denotes the average scale on the rescaled frames (by 50%).

source \ target	YOLOv2	YOLOv3	FasterRCNN	MaskRCNN	RetinaNet [5]	Cascade MaskRCNN [1]
YOLOv2	0.743	0.526	0.000	0.000	0.182	0.000
YOLOv3	0.518	0.701	0.014	0.037	0.453	0.009
FasterRCNN	0.617	0.237	0.930	0.848	0.900	0.695
MaskRCNN	0.547	0.359	0.873	0.855	0.838	0.575

Table S3. The mASRs of transferred attack.

[2] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 5

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 4

[4] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019. 2

[5] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 6

[6] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 5

[7] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 4

[8] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 4

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 4

[10] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In *European Conference on Computer Vision*, pages 665–681. Springer, 2020. 4