Supplementary Material for Infrared Invisible Clothing: Hiding from Infrared Detectors at Multiple Angles in Real World

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

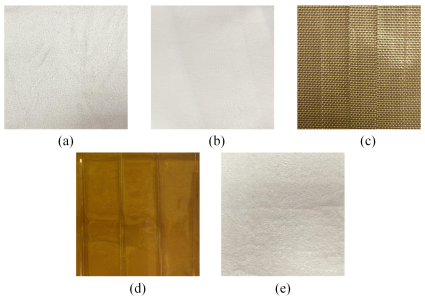


Figure S1. Different materials. (a) Cotton (b) Polyester (c) Teflon (d) Polyimide (e) Aerogel.

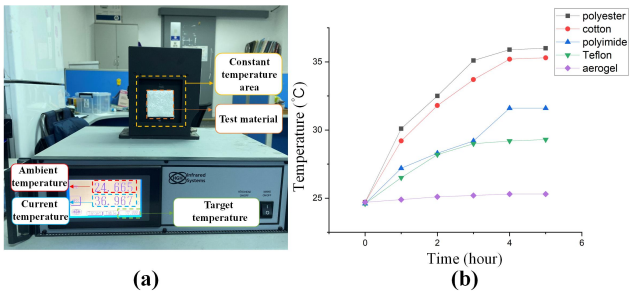


Figure S2. Physical test of thermal insulation materials. (a) Standard heat source (b) Temperature vs. time curve of different materials

1. Adversarial Attack Demo in the Physical World

See *Supplementary Video 1* for the demo.

2. Details about Physical Test of Thermal Insulation Materials

We tested two common fabrics (cotton and polyester), two thermal insulation tapes (Teflon and polyimide), and a new type of material (aerogel). Their photos are shown in Figure S1 . The size of these material samples was $6cm \times 6cm \times 6mm$. We put these samples on the standard heat source (Figure S2(a)) produced by HGH company, which could maintain a uniform temperature in an area of $10cm \times 10cm$ with an accuracy of one thousandth of a degree Celsius. The environment temperature was 24.7°C. The target temperature of the heat source was set to 37.0°C, which is also the temperature of the human body. Then we covered the surface of the heat source with different insulation materials. We used the point temperature measurement function in FLIR T630sc infrared camera to measure the temperature of the material surface. We randomly selected 10 points each time, and calculated the average value as the measured temperature. The distance between the infrared camera and the surface of the material was $30cm$. We measured it every 1 hour, for a total of 5 hours, and obtained results in Figure S2(b). The results showed that the aerogel had good thermal insulation properties and remained stable over time.

3. Hyper-parameter Settings during Optimization

We describe the digital attack process for YOLOv3 [12] in Section 4.3.1 of the main paper. Some hyper-parameter settings were as follows. The batch size was 8, and the initial learning rate was set to 0.03. If the loss function did not drop after 50 iterations, then the learning rate gradually

*Corresponding author.

Figure S3. The manufacturing process of the "QR code" clothing. See *Supplementary Video 2* for more details.

decreased until 0.001, and the optimization process ended at that time. We got a stable pattern in this way. The experiments were conducted on 8 1080Ti GPUs.

4. The Manufacturing Process of the "QR Code" Clothing

Figure S3 describes the manufacturing process of the "QR code" clothing. See *Supplementary Video 2* for more details. We first printed the texture pattern we designed on a $1.5m \times 1.5m$ cloth. The purpose is to better guide us in the subsequent pasting of aerogel materials. We hired a tailor to make the cloth into a piece of clothing. Next, we stuck double-sided tape on the black area of clothes. Then, we cropped the aerogel felt into many blocks and encapsulated the blocks with scotch tape, and finally glued the blocks to the black area of the cloth.

5. Details about Ensemble Attack

During the optimization process of the ensemble attack, we used four models: YOLOv2 [11], YOLOv3, Faster-RCNN [13], Mask-RCNN [4]. The YOLOv2 model was implemented in adversarial-yolo [1]. The YOLOv3 model was implemented in Pytorch-YOLOv3 [7]. The Faster-RCNN and Mask-RCNN model were implemented in the torchvision [10] library. We used the Adam optimizer [5]. The batch size was 1. The settings of learning rate were the same as described in Section 3 in the *Supplementary Material*. During the attack process, We tested three models: RetinaNet [6], Libra-RCNN [9], and Deformable DETR [17]. They were all implemented in the mmdetection [8] library.

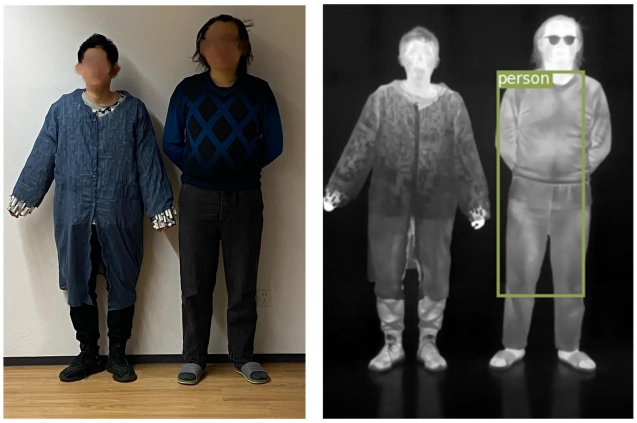
6. Details about Adversarial Defense Methods

We tested five typical methods to defense our attack method in the digital world. These methods included pre-processing defenses (spatial smoothing [16] and Total Variance Minimization [3]), adversarial training [2], and their combinations (adversarial training + spatial smoothing and adversarial training + Total Variance Minimization). For spatial smoothing, we chose a typical method: Gaussian filtering. We used the Gaussian filter implemented in the SciPy [14] Ndimage library. For Total Variance Minimization, we used the Total Variance Minimization module in Adversarial Robustness Toolbox [15] library. For adversarial training, we used the data augmentation method. The adversarial training data contains clean images and adversarial images, and the ratio of adversarial images to clean images was 1:5. Other training settings were the same as in Section 4.3.1 of the main paper. We used the adversarial texture shown in Figure 4(a) (in the main paper) to test the effectiveness of these defense methods. The results are shown in Table S1.

The results indicated that these methods had a certain de-

Table S1. Defense Methods

Defense Methods	AP increase by
Gaussian Filtering	18.7%
Total Variance Minimization	17.9%
Adversarial Training	22.1%
Gaussian Filtering + Adversarial Training	24.5%
Total Variance Minimization + Adversarial Training	22.3%



(a) (b)

Figure S4. A person wearing adversarial clothing with a cover (left) and a person wearing ordinary clothing (right). (a) Visible light view. We blurred facial area for privacy reasons. (b) Infrared view.

fense effect, and the combination methods were better than the single defense method. The most effective way (adversarial training + Gaussian filtering) increased the AP from 12.3% to 36.8% only, and our attack method still lowered the AP by 63.2%, which showed that our attack method was still effective.

7. Adversarial Clothing with a Cover

In the previous experiments, we stuck the aerogel blocks on the surface of the clothes. But this type of clothing may look strange under visible light view. To make the adversarial clothes more natural in the visible light view, we put an extra layer of ordinary clothing on the adversarial "QR code" clothing (Figure 7(b) in the main paper). Figure S4(a) shows a person wearing adversarial clothing with a cover (left) and a person wearing ordinary clothing (right) in the visible light view. We photographed them with visible light and infrared cameras respectively, and input the infrared images into the YOLOv3 model. The detection threshold was 0.7, same as before. Figure S4(b) shows an example. Although the adversarial "QR code" pattern looked a little blurry than the case of wearing only adversarial clothes, the adversarial attack could still be successful because of the penetrability of thermal infrared imaging. This inspired us to use aerogel as the interlayer of clothes. One interesting extension is to make adversarial clothing that looks like a down jacket, which will have both adversarial effect and warmth retention effect.

References

[1] EAVISE. Adversarial yolo. [EB/OL]. https://gitlab.com/EAVISE/adversarial-yolo/ Accessed Nov. 21, 2021. 2

[2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 2

[3] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. In *ICLR*, 2018. 2

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015. 2

[6] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 2

[7] Erik Linder-Norén. Pytorch-yolov3. [EB/OL]. https://github.com/eriklindernoren/PyTorch-YOLOv3 Accessed Nov. 21, 2021. 2

[8] OpenMMLab. Mmdetection. [EB/OL]. https://github.com/open-mmlab/mmdetection Accessed Nov. 21, 2021. 2

[9] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *CVPR*, 2019. 2

[10] Pytorch. Torchvision. [EB/OL]. https://github.com/pytorch/vision Accessed Nov. 21, 2021. 2

[11] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *CVPR*, 2017. 2

[12] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 1

[13] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017. 2

[14] SciPy. Scipy. [EB/OL]. https://github.com/scipy/scipy Accessed Nov. 21, 2021. 2

[15] Trusted-AI. Adversarial robustness toolbox (art) v1.8. [EB/OL]. https://github.com/Trusted-AI/adversarial-robustness-toolbox/ Accessed Nov. 21, 2021. 2

[16] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS*, 2018. 2

[17] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *ICLR*, 2021. 2