Journal of Information Security and Applications 69 (2022) 103278

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

A motional but temporally consistent physical video examples

Zhenyu Du [a], Xingxing Wei [b], Weiming Zhang [c,*], Fangzheng Liu [a], Huanyu Bian [c], Jiayang Liu [c]

[a] College of Electronic Engineering, National University of Defense Technology, Hefei 230037, China
[b] Beihang University, Beijing 100000, China
[c] University of Science and Technology of China, Hefei 230037, China

ARTICLE INFO

Keywords:
Adversarial examples (AEs)
Action recognition
Spatial motional
Temporally consistent

ABSTRACT

Adversarial examples (AEs) attract extensive attention due to their inherent security-related properties of attacking Deep Neural Networks (DNNs) through carefully constructed modifications. Recently, they have been extended to video tasks. Human action recognition based on DNNs is a crucial task in video tasks. AEs of human action recognition models attract much focus and previous works demonstrated the vulnerability of AEs of human action recognition in the digital world. However, the adversarial videos for attacking human action recognition models in the physical world are still in the open stage. The design of physical adversarial videos is crucial for helping to evaluate the robustness of some critical applications based on human action recognition, e.g., surveillance and pedestrian detection. Unlike the digital attacks on action recognition models, the perturbations of physical adversarial videos should be motional but temporally consistent across each whole video. The attacks need to destroy the spatial interactions and temporal interactions in videos. Previously developed attacks for video models in the digital world are difficult to transfer to the physical world. In this paper, we close this gap, and we are the first to attack human action recognition models in the physical world. We first generate a dynamic mask via an improved object tracking method and then use the center location to construct a motion location map. Finally, gradient sharing method is used to generate temporally consistent perturbations and optimize the perturbations into robust patches. Experiments show that these patches can successfully attack a real-time human action recognition system, and the proposed approach has a 77.5% success rate in this setting.

1. Introduction

Adversarial examples (AEs) attract extensive attention due to their inherent security-related properties. Recently, researchers have found that adding subtle perturbations to the input of deep neural networks causes models to give a wrong output with high confidence. Furthermore, they call the deliberately constructed inputs adversarial examples (AEs). The attack of DNNs by AEs is called adversarial attacks. These low-cost adversarial attacks can severely damage applications based on DNNs.

Current research of AEs can be divided into two main categories: one is digital AEs, and the other is physical AEs. And compared with digital AEs, physical AEs bring more harm to reality. Adding adversarial patches onto traffic signs can lead to auto-driving system error [1]. Adding adversarial logos to the surface of goods can impede automatic check-out in automated retail [2]. Generating adversarial master prints can destroy deep fingerprint identification models [3]. In any scenarios mentioned above, AEs can cause great inconvenience and harm people's lives. Therefore, physical AEs has become an urgent issue in AI security.

Action recognition based on deep neural networks (DNNs) is a key task in computer vision. The AI systems-based on action recognition models are widely used in the field of smart homes [4], automatic driving [5], elderly care and property protection [6]. However, DNNs are vulnerable to adversarial examples [7]. Recent works have shown that an image with small perturbations can fool a classification system trained by DNNs. Such images with imperceptible perturbations are called adversarial samples. Recently, researchers have expanded the scope of DNNs to action recognition [8–12].

Since most applications based on action recognition systems are directly related to personal safety and property security, it is crucial to study the adversarial examples faced by video models in the physical world. Two common scenarios are encountered. **(1) Intelligent security attacks.** In an intelligent protection environment, many intelligent security cameras used by a smart city can recognize special behaviors according to videos. In some important places such as government agencies, it may use intelligent cameras to detect harmful actions and

* Corresponding author.
E-mail addresses: dzy17@nudt.edu.cn (Z. Du), xxwei@buaa.edu.cn (X. Wei), zhangwm@ustc.edu.cn (W. Zhang), yoyofangzheng@aliyun.com (F. Liu), hybian@mail.ustc.edu.cn (H. Bian), ljyljy@mail.ustc.edu.cn (J. Liu).

https://doi.org/10.1016/j.jisa.2022.103278

Available online 19 July 2022

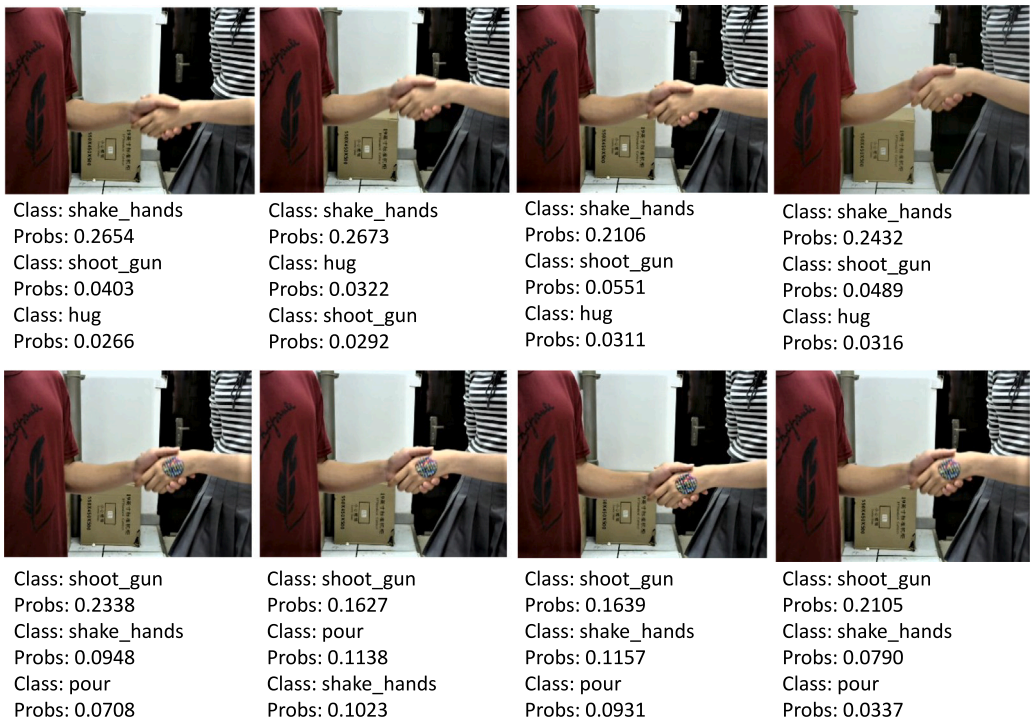


Fig. 1. This figure shows the physical attack results of real-time action recognition models. The first line contains the adversarial videos, and the second line includes the original videos after adding an adversarial patch to the palm. The 'shake-hands' action can be mistaken for 'shoot-gun'. The radius set in this experiment is 50 pixels. At the bottom of each image is the label of the current frame.

automatically trigger an alarm. When some attackers utilize elaborate adversarial patches accidentally, they can hide their harmful actions or make the cameras misclassify normal action into other harmful actions, such as recognizing the action 'shaking hands' into 'shoot gun'. **(2) Smart home attack.** At present, more and more families use full coverage smart homes to control intelligent devices. Some smart homes use human gestures to convey instructions to devices. If an attacker uses a specially crafted adversarial patch to make these smart homes mistakenly recognize the owner's instructions, it will bring additional damage. For example, suppose the home equipment recognizes the initially set "wave" action as other meaningless actions, such as "smiling". In that case, it will make the instructions unable to be conveyed correctly and make the equipment ineffective.

These aforementioned scenarios shows the physical AEs of action recognition systems can cause great inconvenience and bring threat to people's lives. However, no related work has attempted to explore this problem and current physical adversarial video examples are still in an open stage. In this paper, we perform the first adversarial video attack, and attempt to attack the human action classifiers under a simple situation to verify the possibility of solving this problem. Fig. 1 shows the physical attack results of the real-time action recognition models.

We summarize our contributions as follows:

- We propose a general framework that attacks the physical world's classifier of action recognition.
- We heuristically propose a method to solve the problem of generating dynamic adversarial perturbations that change with the temporal frames but maintain consistent values.
- We perform experiments to verify our adversarial perturbations (a real patch) in the digital world and the physical world with different experimental settings. The proposed approach has a 77.5% fooling rate in the physical setting and a 100% rate in the digital world.

The rest of this paper is organized as follows. In Section 2, we first analyze the differences between physical AEs and digital AEs and give the main problem of using digital AEs in the physical scenarios. In , we briefly review the related work from three aspects, including digital AEs in images, digital AEs in videos and physical AEs in images. In , we give the details of our algorithm. In Section 5, we present our experimental results and compare them with other baseline methods. Finally, we conclude our paper in Section 7. And we also give the analysis of the Shoelace Theorem, which we introduced in Appendix.

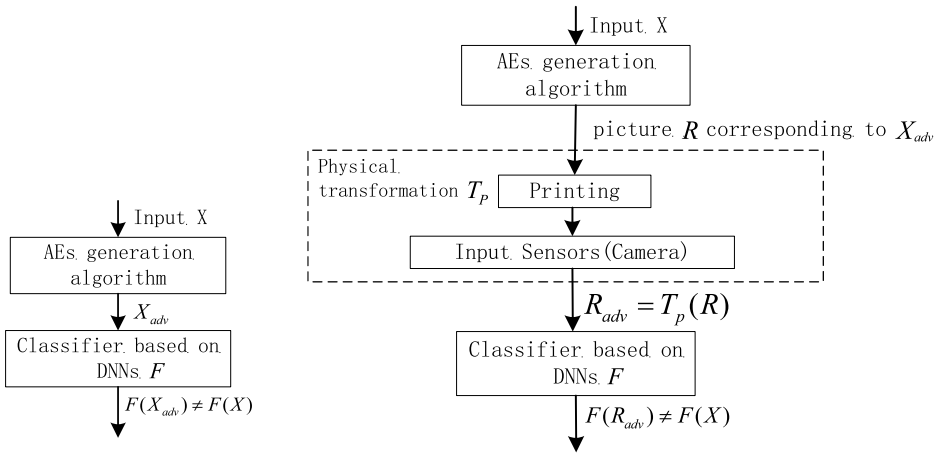
2. Analysis of physical AEs of action recognition

To generate physical adversarial video examples, we first analyze the differences between physical AEs and digital AEs and give the main problem of using digital AEs in the physical scenarios. Due to these problems, it is important to generate physical AEs that can overcome those vulnerabilities. We thus analyze the features of physical AEs. And then we sum up the specific features of physical adversarial video examples according to the features of physical AEs. Finally, we conclude that the main features of generating adversarial video examples are the preservation of the motion of adversarial perturbations in spatial domain and consistency in the time domain.

First, we analyze the differences between physical AEs and digital AEs and give the main problem of using digital AEs in the physical scenarios.

Current research of digital AEs is based on the assumption that the data of AEs can be directly input into the DNNs. Fig. 2 compares that difference.

The left flow-chart in Fig. 2(a) shows the generation process of the digital AEs. When X is fed into the AEs generation algorithm, it outputs the adversarial example X_{adv} and then X_{adv} is directly inputted into the model classifier F, causing the model to make a wrong classification. The right flow-chart in Fig. 2(b) is the generation process of the physical AEs. When X is fed into the AEs generation algorithm, it outputs the adversarial example X_{adv} and its related picture R. And then, after the physical transformation Tp including printing, taking a photo, etc., it finally enters R_{adv} into the classifier F and makes it misclassify that R_{adv}.



(a) Flow-chart of digital AEs generation

(b) Flow-chart of physical AEs generation

Fig. 2. Comparison of digital and physical flow.

The assumption of inputting data into the model without being physically transformed does not hold in many physical scenarios, such as video surveillance systems, robots that input data through cameras, and image-based mobile applications. When generating physical AEs, this assumption leads to the main problems: the AEs generated under the assumption would lose their antagonism when the physical environment changes. The secondary input of physical transformation will add additional noise and change the pixel value of generated AEs, making the AEs lose their antagonism. Specifically, due to the change of viewing angle, distance, and illumination, the adversarial perturbations generated by a single specific constraint would lose their antagonism under other environmental conditions.

In addition, the digital AEs cannot attack the real-time classifiers. The physical scenarios are usually real-time models, but digital AEs need to be pre-processed and then input the models to complete attack. It means that when given a clean input X and a threat model F, in order to complete attack, the attackers must pre-process X into X_{adv} and then re-input it to the model. But in physical scenarios, it has no time to complete the process of pre-treatment. In physical scenarios, it needs to generate the general adversarial perturbations in advance, such as a patch P, and put it on the objects directly to complete the attacks.

Moreover, the digital adversarial perturbations of digital AEs are difficult to implement in physical scenarios. The digital AEs generation algorithm has no limit on the generation space of the adversarial perturbations and is usually added to the background of the image. However, in practice, the background change is challenging and limited to space–time constraints leading to the perturbations that cannot be realized on the physical entity.

To overcome these problems, physical AEs should have the features as follows. (1) Physical AEs need to be robust to environmental changes. Physical AEs should overcome the disturbances to the adversarial features from secondary inputs as mentioned in Fig. 2(b). (2) Physical AEs need have the capability of attacking real-time models. (3) Perturbations of physical AEs need to be placed on physical entities. The adversarial perturbations generated by traditional digital AEs generation algorithms are scattered in the whole input. Taking an image as an example, most of the adversarial perturbations appear in the picture's background. However, changing the pixels of the background environment is challenging in the real physical world. Therefore, digital AEs cannot adapt to the physical world, and perturbations of physical AEs need to be limited to physical entities.

As Fig. 3 shows, the left is the AEs generated by the digital AEs generation algorithm, and the right is the physical AEs. The perturbations of digital AEs in the left are scattered in the whole input, including the foreground(panda) and other background pixels. The perturbations of physical AEs on the right are limited only to physical entities in the foreground (traffic sign).. The perturbations on the right are easily added in the physical world. But the perturbations of the left in the background are difficult to realize in the physical world. Therefore, the perturbations of physical AEs need to be placed on the physical entities.

Therefore, **physical AEs are the adversarial samples that can be placed on physical entities and can overcome the problem of adversarial failure caused by secondary input and environmental changes.**

Moreover, we then analyze the features of physical adversarial video examples according to the features of physical AEs.

(a) The first feature is the AEs can be placed on physical entities. When considering the physical adversarial video examples, the physical entities are motional thus the generation algorithms should ensure adversarial perturbations keep spatial motional and temporally consistent. The analysis are as follows.

The perturbations added to physical adversarial videos should be spatial motional. It means that the perturbations should follow the motion of moving objects. Previous works completed adversarial attacks with immobile perturbations in the physical world [2,14,15]. They select stationary foreground objects to add perturbations. But when it turns to dynamic action recognition tasks, the foreground (objects or persons) and most backgrounds are motion with the video playing. Thus, the immobile adversarial perturbations may disappear from the video in the following frames so that they lose adversarial capability. Meanwhile, it is hard to realize the changed perturbations on the motional background in the physical world. Therefore, the best method is to keep the perturbations also moving with the foreground, i.e., the moving physical entities.

Furthermore, the value of perturbations should keep temporally consistent during video playing. Traditional adversarial attack method of digital adversarial video examples not only dense-attack but sparse-attack are temporal inconsistent. They focus on selecting all frames (dense-attack) or several frames (sparse-attack) to add perturbations and pay no attention to limiting the value of perturbations to keep equal in each perturbed frame, which causes temporal inconsistent perturbations. But that temporal inconsistent perturbations cannot be realized in the physical world. We cannot manually realize the dynamic perturbations that change with the fast frame conversion rate of the digital adversarial videos. Even if machine can realize them automatically, the perturbations may lose interference capability because of the different framing rates of different models and different devices [16].

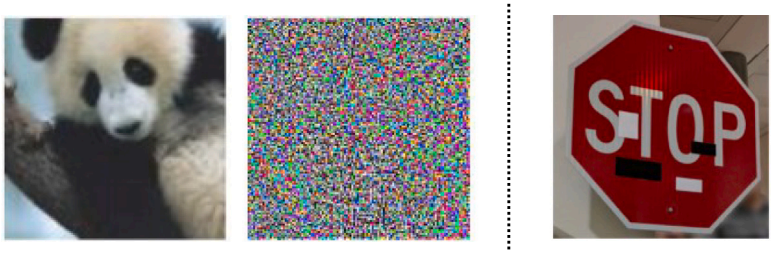


Fig. 3. Figure for the analysis of physical perturbations should be placed on physical entities. *Image Credit:* left Goodfellow et al. [7], right Eykholt et al. [13].

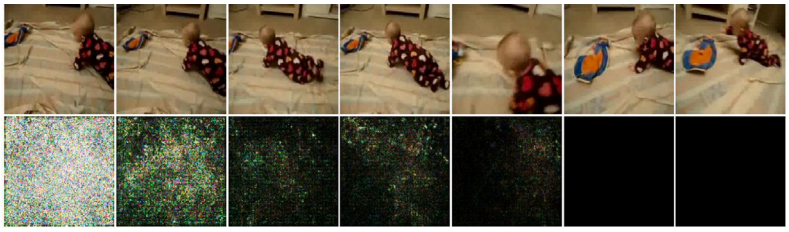


Fig. 4. Figure for the analysis of physical perturbations should keep temporally consistent. *Image Credit:* Wei et al. [17].

Fig. 5. Figure for the analysis of physical perturbations should robust to speed and distance changes.

Thus, the better method to solve the difficulty of controlling the various perturbations during video playing is to maintain the perturbations consistent during a video. As Fig. 4 shows, the perturbations in different frames have different value. Some perturbations appear in a frame and disappear in the following frames. It is difficult to be realized in the physical world due to the fast framing rates. Therefore, the physical perturbations should keep temporally consistent.

(b) The second feature is that the AEs can overcome the problem of adversarial failure caused by secondary input and environmental changes. For the physical adversarial video examples, the main environmental changes they face are the changes in distance and speed. The changes in distance between the moving objects and input devices lead to changes in the space proportion of the adversarial perturbations in the whole sample. It means that the larger the distance, the smaller the space proportion. Then it has a significant impact on the antagonism of AEs. When the distance is farther, the antagonism will be weakened. The changes in the speed of moving objects are another environmental change factor that significantly impacts the antagonism. When the speed is too fast, it is difficult for the input device to capture a clear image of AEs, significantly impacting the clarity of the adversarial perturbations and then influencing the antagonism. As Fig. 5 shows, the left images show the difference of the space proportion of the adversarial perturbations in the whole sample of different distance. The right images show the difference of clarity of the adversarial perturbations of different speed. The changes of distance and speed have an impact on the antagonism of AEs.

Moreover, the AEs should be robust to the secondary input. The main difference for the physical adversarial video examples is that different input devices have different framing styles. Especially, they may frame the first 16 frames or the last 16 frames as the input of models.

Therefore, **the key to generate adversarial video examples is to keep the adversarial perturbations spatial motional and temporal consistent and to make them robust to the changes of distance, speed and framing style.**

In this paper, we propose a new type of adversarial attack that generates the motional but temporal consistency physical adversarial perturbations (a real patch) to deceive the classifier of video classification. Specially, we generate the precise dynamic masks that can change the location with the moving objects to keep the perturbations motional. And we propose the method of gradient drift to make the value of perturbations keep consistent. And we use transform function to make the perturbations robust to the changes of distance, speed and framing style.

3. Related work

3.1. Digital adversarial attack against image models

In recent years, many works focused on generating adversarial samples for images. These works can be divided into two categories due to their attacking condition: white-box and black-box attacks.

White-box attack assumes that the structure and parameters of a target model are known to the attacker. In white-box attack, it is easy to perform attacks according to the gradient of objective function of attack or the forward derivative of model computed via back propagation. Various of attacks have been proposed. For example, L-BFGS [18], Fast Gradient Sign Method (FGSM) [7], Deep Fool [5], Jacobian-based Saliency Map Attack (JSMA) [19], Basic Iterative Method (BIM) [20], C&W attack [21] and Universal attack [22].

Contrarily, the black-box setting remains the model information unknown to attacker, which makes generation of adversarial samples more challenging. Existing works in this setting include Zeroth Order Optimization (ZOO) [23], Autoencoder-based Zeroth Order Optimization Method (AutoZOOM) [24], decision based attack [25], Opt-attack [26].

3.2. Digital adversarial attack against video models

Then some works extend adversarial image examples into video examples, they focused on generating adversarial digital videos to act against DNNs-based classifiers. The first sparse attack [17] to be proposed for video models used the $l_{2,1}$-norm regularization-based optimization method. After that, Jiang et al. [27], Wei et al. [28] utilized the gradients to generate perturbations on selected frames, maintaining a low level of sparsity. Li et al. [16] studied a framing method for a real-time action recognition system and attacked it by polluting the selected frames. For dense attacks, Zhang et al. [29] proposed generating adversarial perturbations by using a motion map, thereby achieving better performance. And Zajac et al. [30] demonstrated the validity of adding perturbations only on the border of each frame.

However, these digital attacks fail to be realized in the physical world. Sparse attacks only add perturbations to several video frames, and it is difficult to control perturbations that exist in one frame but disappear in the next frame. Dense attacks are devoted to generating imperceptible perturbations, perturbations with varied positions and values [29], or perturbations with fixed positions but various values [30]. Specifically, the imperceptible perturbations' values are usually small to achieve better invisibility. Therefore, they lose antagonism easier under the secondary input of the physical world. Meanwhile, the perturbations with varied positions and values and those with fixed positions but various values do not correspond to the features of adversarial videos against action recognition models. Effective adversarial videos need to have features that are spatial motional and temporally consistent, as we concluded from the analysis we did earlier. Therefore, those digital adversarial perturbations cannot be applied in the physical world.

3.3. Physical adversarial attack against DNNs

Recent works have attempted to generate adversarial examples to act against the model of face recognition, object detection, and automatic driving models in the physical world. The works [31,32] developed patches that could be added to real eyeglasses and hats, respectively, to deceive face recognition models. The work [2] proposed a method to generate a universal adversarial patch to attack an image classifier. Other works have focused on attacking object detectors. The work [15] made a person disappear from the person detector by holding cardboard. Xu et al. [33] printed patch on T-shirt to generate adversarial T-shirt against single detectors and multiple detectors. The work [14] made all existing objects in the image miss entirely by placing the patch anywhere in the image. Xu et al. [34] attacked the person detector and makes the person disappear from the detector. And Nassi et al. [35] inserted several abnormal frames in daily billboards to stop Tesla's autopilot car. Eykholt et al. [13] added the patch, which acted as 'graffiti' on a stop sign, to attack an automatic driving model.

However, to the best of our knowledge, no work has attempted to attack action recognition models in the physical world. Video recognition models use videos as input, so the the above methods which take the image-based classifier as the target do not work well because they cannot process the inter-frames relationship. Furthermore, except Xu et al. [33], all the other works mentioned above focused on attacking static objects, e.g., eyeglasses and stop signs. The perturbations changes discussed by those works are small-scale changes, e.g., illumination, angle, and distance changes, which can be implemented by affine transformations. However, those of adversarial videos are large-scale changes, including action and speed changes. The method of attacking static objects fails to adapt to attack moving objects. Although the work Xu et al. [33] attacks motional people, it concentrates on the deformation effects of non-rigid objects (T-shirts), which may not be suitable for that task.

The key to attacking the video recognition models is to generate spatial moving perturbations that change their locations with those of dynamic objects and with temporal consistencies that maintain constant values during video playback.

4. Methodology

As the analysis in Section 2, the perturbations added to adversarial videos should be spatial motional and temporally consistent. The two constraints restrict the generation region and the value of the perturbations, respectively. Specifically, the perturbations need to be dynamic, move with the observed objects and maintaining theirs value consistent during video playback. In the spatial domain, it restricts the adversarial perturbations that should be added to the region of moving objects. In the temporal domain, it restricts the adversarial perturbations' values to be the same in each frame of video.

The framework of our method is shown in Fig. 6. It is divided into two parts: simulation training and physical attacking. In the simulation training, we first sample video frames taken by a real-time camera under our physical setting. Then, we generate dynamic masks based on these frames. And we construct a location map based on the masks. We use these dynamic masks to limit the generation region of perturbations. After that, we initially generate temporally consistent noise, and use the location map to index the motional pixels. And we then use gradient accumulation and sharing technology to let the pixels that shared the same location in the motional frames have the same gradient. Last, we optimize the perturbations via the output of the target model and the physical transform function. In physical attacking, we perform the physical attack by sticking the printed adversarial patch on the moving objects against the real-time human action recognition system.

In the following sections, we detail the methods of the whole framework. It is dived into three parts: (1) generating spatial motional perturbations, (2) generating temporally consistent perturbations and (3) optimizing physical adversarial perturbations.

4.1. Generating spatial motional perturbations

The key to generating spatial motional perturbations is generating dynamic masks. In this subsection, we are motivated by the traditional methods of limiting the perturbed region and then analyze the defect of that method. Finally, we give the detail of our improved method for the same purpose.

Traditional methods use masks of images to control the location of perturbations in the physical adversarial images generation. They generate the masks of images to wipe off some pixels which do not need to be perturbed. Specifically, the masks are the images whose pixels' values are equal to 0 or 1. And the pixels whose value is equal to 1 are the perturbed space, and they are set manually before generating AEs. But the pixels whose value is equal to 0 are the space that does not add perturbations.

However, these static masks are not suitable for physical adversarial video attacks. It is impractical to set each mask of video frames manually in advance due to the high dimension of videos. Therefore, in this paper, we propose automatically generating dynamic masks to control the generation region of perturbations.

The main idea of generating a dynamic mask is to track a moving object in a videos in advance and then draw a circle with pixels' value equal to "255" on a black image according to the tracked object's center and fixed radius. In this paper, we select two actions: "waving" and "shaking hands" as examples to verify the effectiveness of our proposed framework.

We first track the moving objects: the waving hands of a video labeled as 'wave' and the shaking hands of a video labeled as 'shaking-hands' of the HMDB-51 dataset [36]. We are inspired by the Zhang et al. [37] and its sourcecode [38]. The algorithm can detect the palm of a person. Fig. 7 shows the results of that algorithm.

As Fig. 7 shows, the zone of the results produced by that algorithm has two drawbacks: (a) there is a deformation, the size and shape of which changed in some video frames, and (b) some frames cannot be detected by the model. In particular, the palms detected by this

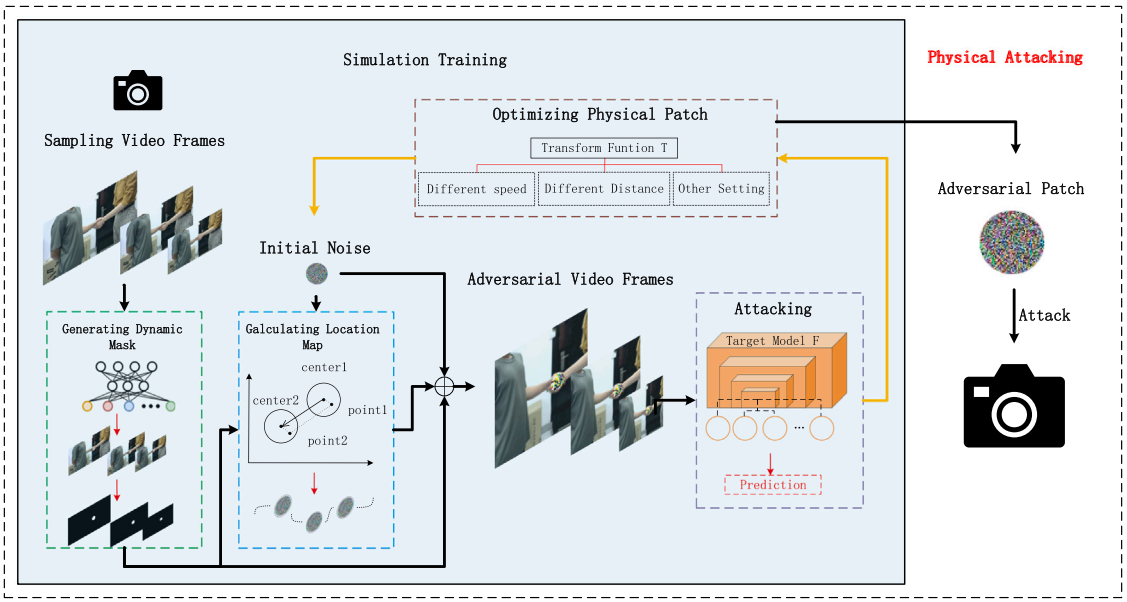


Fig. 6. The pipeline of the proposed algorithm. It is divided into two parts: simulation training and physical attacking.

Fig. 7. Figure displaying the results of Zhang et al. [37]. The video data is from HMDB51 [36]. Deformations and omissions exist in some video frames.

algorithm in the left image and the middle image plotted with black lines have different shapes and positions. And the right image detects no palms.

However, the dynamic masks should meet the characteristics that the sizes and shapes of masks with the pixel values of 1 or 255 need to be constants in different frames. Therefore, to generate the dynamic masks with unified shapes and sizes, we improve the algorithm by two tricks: (a) the center positioning method, and (b) the smooth positioning method.

In the center positioning method, we utilize the key points calculated from the results of the algorithm to calculate the center of the palm via Eq. (1). We set the coordinates of the center of palm as (Ξ, Ψ) and ξ_i defines the coordinate of the x axis of the i_{th} key point, while ψ_i defines that of the y axis.

$$\begin{aligned} \Xi &= \frac{\sum_{i=0}^{n-1}(\xi_i\psi_{i+1} - \xi_{i+1}\psi_i)(\xi_i + \xi_{i+1})}{3\sum_{i=0}^{n-1}(\xi_i\psi_{i+1} - \xi_{i+1}\psi_i)} \\ \Psi &= \frac{\sum_{i=0}^{n-1}(\xi_i\psi_{i+1} - \xi_{i+1}\psi_i)(\psi_i + \psi_{i+1})}{3\sum_{i=0}^{n-1}(\xi_i\psi_{i+1} - \xi_{i+1}\psi_i)} \end{aligned} \tag{1}$$

The key points we select are based on the points of hands tracked by algorithm [37]. We select six points as the key points that label the palm od hands. Fig. 8 shows the points tracked by algorithm [37] in left and we select the [0,1,5,9,13,17] as the key points. And then we introduce Shoelace Theorem to calculate the centroid of the polygon covered by the key points. The Shoelace Theorem is detail in Appendix.

In the smooth positioning method, if a few frames cannot be detected and theirs centers are lost, we can approximately regard the two adjacent center points as linear motions because the displacement between two adjacent frames is tiny. Therefore, we calculate the average value of the former frame's center and the next frame's center as the center coordinate that the algorithm cannot detect. However, in some cases, due to the fast speed of an object, the object cannot be seen clearly. The edge information of the object is vague, so most frames lose the essential data of the center coordinate. The approximation that depends on the adjacent frames does not work.

Therefore, we should model the track of the center's motion. We approximate the motion as a simple linear translation ('shaking-hands') and circular motion ('wave') in our setting. In the shaking hands action, the center linearly shifts back and forth between fixed points. Therefore, the only knowledge we need to know is the coordinates of the fixed points and the motion speed. In wave action, the center can be approximately regarded as exhibiting circular motion with the radius of the 'arm length' and the center of the 'elbow joint'. Therefore, the knowledge that we should acquire includes the coordinates of the 'elbow joint', the 'arms length', and the waving speed. Similarly, although we only take these two actions as examples of our framework, the pipeline of generate adversarial video examples to act against human action recognition models can generalize to other actions.

Fig. 9 shows the dynamic masks generated by our method.

4.2. Generating temporally consistent perturbations

The main idea of generating temporally consistent perturbations is to let the pixels in the dynamic masks of different frames have the same gradients. As Fig. 10 shows, the points A, B, C, D of different frames located in the same position of the palm and theirs perturbations should

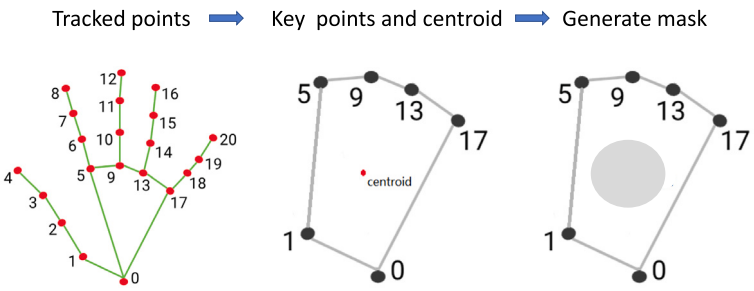


Fig. 8. Figure for tracked points of algorithm [37] (left), the key point selected of our algorithm (middle) and the location mask (right). *Image Credit:* left Zhang et al. [37].

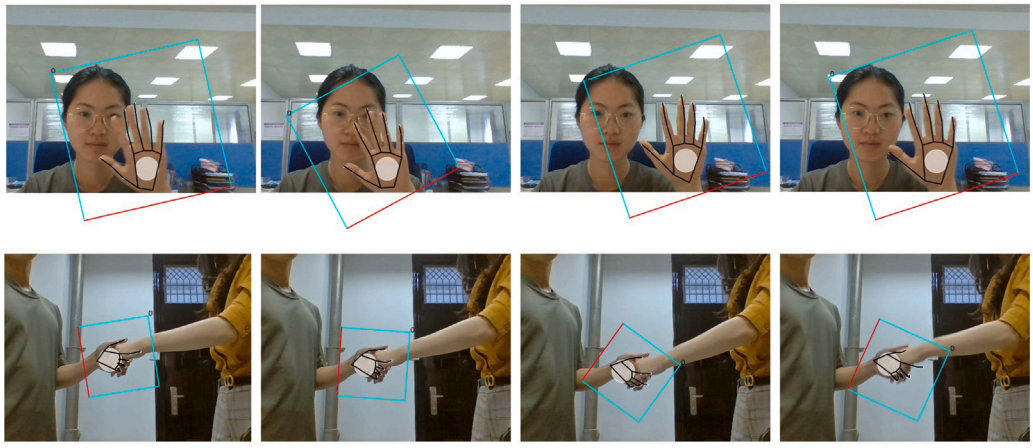


Fig. 9. Figure for the tracking of the moving hands. The first line shows the different video frames of waving hands, and the second line shows the different frames of shaking hands. The radii of the circles in the palms are 40 and 60 pixels, respectively.

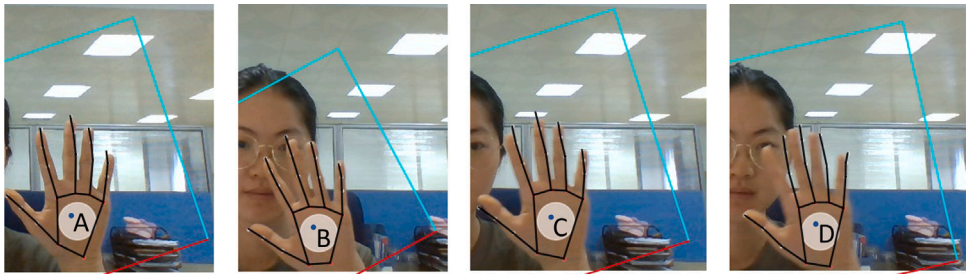


Fig. 10. Analysis of the temporally consistent perturbations. The points A, B, C, D of different frames are located in the same position of the palm, and theirs perturbations must remain consistent.

keep consistent. In this subsection, in order to achieve this goal, we are motivated by the traditional method of generating adversarial images' perturbations and then analyze the defect of using it in generating adversarial videos' perturbations. Finally, we give the improved method under our framework.

The traditional method of generating images' perturbations, called the gradient-based AEs generation method, usually uses the gradients of inputs to update the perturbations. The equation of these methods is as follows.

$$X_{adv} = X + \alpha \cdot \text{sign}\left(\nabla_X \mathcal{J}(F(X))\right) \tag{2}$$

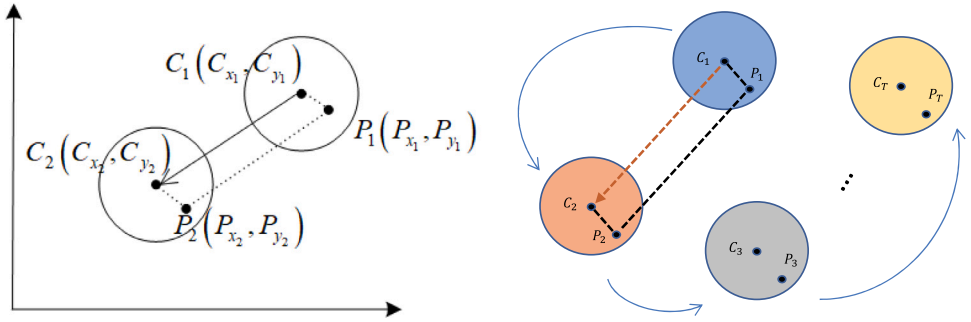
In Eq. (2), the $\nabla_X \mathcal{J}(F(X))$ defines the gradient of input X under the loss function $\mathcal{J}(F(X))$.

However, the adversarial perturbations generated by this method of AEs generation are global. Using this method to generate adversarial videos will add perturbations to the whole video. Moreover, that method adds adversarial perturbations to the pixels based on their gradients, and the gradients' values and direction of pixels in different positions in the video are inconsistent. Therefore, if using this method to generate adversarial video examples, the adversarial perturbations added on pixels at different parts in each frame have different values. We use the points A, B, C, D in Fig. 10 as examples to explain that conclusion. As Fig. 10 shows, the points A, B, C, D belong to the same region of the palm in different frames, but they have different positions in the image. Therefore, those pixels have different gradients so that they will get different perturbations if using Eq. (2) to generate adversarial videos.

In this subsection, we design a motion map and devise a pixels' gradients accumulation and sharing method. These methods can ensure that adversarial perturbations belonging to the same region of motion tracking in each frame keep consistent. Taking A, B, C, D in Fig. 10 as an example, i.e., the gradients of these points remain constant.

(1) Generating motion map.

The motion map help to locate the moving pixels in different frames. It obtains the position migration map of the same pixels belonging to the same motion track in different frames.



(a) The calculation of motion map (b) Motion map

Fig. 11. Figure for the motion map and its calculation. C_1 and C_2 are the centers of two circle. We can calculate the coordinate of P_1 based on Parallelogram rule and P_2 in left. And after all the pixels have been calculated, it construct a motion map from C_1, C_2 to C_T. Suppose the video has T frames.

As Fig. 11(a) shows, we can calculate any points' coordinates in the motion map. Specifically, given the center coordinates of the dynamic masks in any two frames of the video $C_1(C_{x_1}, C_{y_1})$ and $C_2(C_{x_2}, C_{y_2})$, and a point in these two frames $P_1(P_{x_1}, P_{y_1})$, its position coordinate P in another frame can be calculated based on Eq. (3). And then, after calculating all the frames, it constructs the motion map and indexes all the motional pixels. Supposing the video has T frames. It is the motion map from C_1, C_2 to C_T.

$$\begin{aligned} P_{x_1} &= P_{x_2} - sign(C_{x_2} - C_{x_1})\left|C_{x_2} - C_{x_1}\right| \\ P_{y_1} &= P_{y_2} - sign(C_{y_2} - C_{y_1})\left|C_{y_2} - C_{y_1}\right| \end{aligned} \tag{3}$$

(2) Gradients accumulation and sharing.

The main idea is to accumulate and share the gradients at the same position of different frames. According to this method, the pixels in the fixed position among the same motion trajectory share the same gradients' directions and sizes. Specifically, we calculate the gradient of inputs and then accumulate their gradients of different frames. It is important that we only accumulate the gradients of the pixels in the same location within the location map and keep other pixels' gradients not changing.

We define the classifier function of the target model as $F(\cdot)$. A clean video $V \in \mathbb{R}^{T\times W\times H}$ is input, where T, W, H denote the number of frames, frame width, and frame height, respectively. The ground-truth label of V is defined as $k^* \in \{1, \ldots, K\}$, where K is the number of classes. $V = \{V_i | i = 1, \ldots, T\}$, $V_i \in \mathbb{R}^{W\times H}$ is the ith frame of V. An adversarial video $V_{adv} \in \mathbb{R}^{T\times W\times H}$ yields $F(V_{adv}) \neq k^*$ in the un-target attack and $F(V_{adv}) = k^t$ in a target attack, where k^t is the target label. Set the gradients of the adversarial V_{adv} is $\mathcal{G}$, which can be calculated via Eq. (4). $\mathcal{J}$ is the cross-entropy loss function.

$$\mathcal{G} = \nabla_{V_{adv}} \mathcal{J}(k^t, F(V_{adv})) \tag{4}$$

$\mathcal{G}^{T\times W\times H}$ is a cube, and each element $g(i, w, h)$ can be calculated by Eq. (5). If the pixels are located in the area of moving objects, the gradient of each pixel should remain consistent, and the total value should be the sum of all the pixels located at the same location in the T frames according to the location map.

$$g(i, w, h) = \begin{cases} \sum_{j=1}^{T} g(i, \gamma_j(w, h)), & if\ P(w, h) \in \gamma_j \\ g(i, w, h), & otherwise \end{cases} \tag{5}$$

We define the motion map as γ, where γ is a dictionary, that stores the correspondence of each motional pixel in different frames. γ_j defines the location map in the j_{th} frame. $\gamma_j(w, h)$ defines the coordinate of pixel $P(w, h)$ of the j_{th} frame.

4.3. Optimizing physical adversarial perturbations

In this subsection, we begin to generate and optimize the defined physical adversarial perturbations. The perturbations added to the videos are denoted as r. The adversarial video V_{adv} can be calculated according to Eq. (6). The adversarial perturbations r will be updated during each iteration. τ is a constant. m is the dynamic masks. The sign $\cdot$ is the tensors' Hadamard product.

$$\begin{aligned} V_{adv} &= V \cdot (1 - m) + \mathcal{G} \cdot r \cdot m \\ r &= r + \tau \times \mathcal{G} \cdot m \end{aligned} \tag{6}$$

The perturbations added to the original video should also be robust in the real world. This means that they should remain adversarial in the physical world. However, many interfering factors can change pixels values of the adversarial perturbations generated in the digital world. Furthermore, they weaken the adversarial features of these perturbations. The factors are as follows.

(a) Printing devices may destroy some pixels, and these pixels cannot be printed accurately.

(b) Different environment settings, e.g., the distance and the speed. Different distances and speeds would impact the proportion and clarity of adversarial perturbations. These different settings weaken the effectiveness of perturbations.

(c) Different input devices may eliminate the generated adversarial frame sequences. For example, the AEs are generated for the first T frames while the machines take the last T frames as input, making the adversarial perturbations lose Antagonism.

First, we eliminate the influence of printing devices. The work [15] defined a nonprintability score (NPS) to measure the distance between a perturbation vector in the digital world and the corresponding vector that can be printed accurately in the physical world. The work [33] proposed modeling a map between the real vectors and the vectors obtained after printing. Regarding camera noise, different cameras introduce different noises, and the noise values are different in different environment settings. With the development of camera devices, the noise introduced by the camera is tiny and may have little influence on the perturbations. To solve the difference between the digital perturbations vectors and those vectors obtained after printing and taking photos in our environmental settings, we model a map between the vectors to solve this difference between the digital perturbations vectors and these vectors. We propose a mapping method to eliminate the difference. We first generate the original color matrix λ with a widely used color, print and take a photo for the set with the camera, and obtain a matrix α. We construct a map (translation function of secondary inputting devices) Λ_c between λ and α. If we want to repair the processed image R to a neighbor of the original image, we can calculate the matrix $\Lambda_c(R)$ via Eq. (7).

$$\Lambda_c(R) = R + (\alpha - \lambda) \tag{7}$$

Second, we eliminate the influence of different speeds and distances. Unlike other static physical perturbations against image classification, object detector, and other DNNs-based models, the perturbations against action recognition models are influenced by the motional

speed of the objects of interest. This means that the perturbations added to physical adversarial videos should be robust to different speeds. We define the translation function of the different speeds as Λ_s. What is more, for eliminating the influence of varying distances, we define the translation function of different distances as Λ_d. Specifically, we use iterative training to weaken the influence of different speeds. We add a large number of data with varying speeds to training data. This iterative training leads to the distribution of input data that can fit those data, and the decision boundary of models is capable of recognizing the data with different speeds. And we use geometric transformation to weaken the influence of various distances. We transform the adversarial perturbations to different sizes, keeping the shape consistent, and then add them to the training data for the same purpose. Last, we generate the hybrid optimized AEs.

Third, we eliminate the influence of different input styles. Using the first T frames to generate adversarial perturbations would weaken the antagonism when adding them on other T frames. The adversarial perturbations cannot be universal of different input styles. We also use iterative training to solve the problem. If a video has full L frames and the input of the model has T frames, we define the translation function of different inputting styles as Λ_l. This function is capable of helping the perturbations remain valid when added to T arbitrary and consecutive frames. Thus, The translation function is detail in Eq. (8).

$$\Lambda_l(V) \rightarrow \forall \mathrm{i}, F(\langle V_i, V_{i+1}, \ldots, V_{i+T} \rangle + r) \neq F(\langle V_i, V_{i+1}, \ldots, V_{i+T} \rangle) \tag{8}$$

Above all, we define the physical transformation function as $\boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}$ contains the transformations of Λ_c, Λ_d, Λ_s and Λ_l. Therefore, the objective function is Eq. (9) and the adversarial examples V_{adv} is updated by its condition.

$$\begin{aligned} &\arg\min_r (F(\mathcal{G} \cdot r \cdot m + V'_{adv} \cdot (1-m), t)) \\ &\text{s.t. } V'_{adv} = \Lambda\left(V_{adv}, \Lambda_c, \Lambda_d, \Lambda_s\right) \end{aligned} \tag{9}$$

5. Experimental results and discussion

We divide the experiment into two parts.

(1) Verifying the initial antagonism of the AEs generated by the algorithm. In this part, the experiment first tests the effectiveness of the AEs generation method for the known dataset, that is, using that method to generate AEs for the video of the digital world and test theirs performance.

(2) Verifying the transformed robustness of the AEs generated algorithm against disturbances from the physical world. In this part, we first generate the approximated dataset in the simulated physical environment. Then we generate the AEs for that dataset in the digital environment, print the adversarial patch, and paste it on the surface of the moving object according to the calculated location. It can keep the objects maintaining the same motion as the trained AEs. Finally, we input it into the model through the input device to test its practical effect.

For convenience, the two parts above will be called the digital world AEs experiment and the physical world AEs experiment, respectively.

5.1. Experimental setting

Datasets.

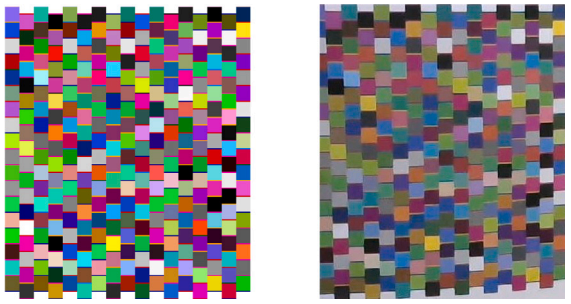
Digital world AEs experimental dataset: HMDB-51 [36]. The dataset contains 51 actions, including 6849 videos of facial activities, body actions, object interaction, and human interaction. The experiment selects two kinds of actions: “shaking hands” and “wave”, and it randomly selects 10 videos of each action as input.

Physical world AEs experimental dataset: this part first pre-records 10 videos of two categories (“shaking hands”, “wave”) under different environmental settings as the training dataset of the physical world AEs. Settings of different environments are “laboratory environment” and “outdoor environment”, and various lighting settings are “LED energy-saving lamp” and “Sunlight”. And the settings of different speeds and distances are shown in the parts of the following experiment. We refer to that dataset as a “sampling dataset” for simplicity.

Table 1
Accuracy of threat nodels under the datasets.

Model	Datasets	
	HMDB-51	Sampling dataset
C3D	59.57%	84.00%
LRCN	37.24%	64.00%
I3D	60.4%	56.00%

Fig. 12. Figure for showing the environment settings of the experiment of physical AEs generation..



(a) Original Image (b) Processed Image

Fig. 13. The difference between original image and the image after printing and taking photo.

Models.

Threat model: regarding the digital world AEs experiment, we select I3D [39], C3D [40] and LRCN [41] as the threat models. In physical world AEs experiment, we transformed those models as the real-time models and then test the performance.

The accuracy of these models on the HMDB-51 dataset and sampling dataset is shown in Table 1. It is noted that the C3D model takes 48 frames as the input, and the I3D model and LRCN model take 16 frames as the input.

Physical Environment Equipments.

As Fig. 12 shows, we place the input device (camera: ‘Logitech-c270’) on display and put the display on the surface of the box on the table to ensure the height of the camera easy to record the actions. And we set a “LED energy-saving lamp” above the camera, keeping the light constant.

In addition, the printing device in the experiment is the “HP Deskjet 1110 series”. Fig. 13 shows the difference between the original picture and the picture after transformation, including print and take photos.

Metrics.

The evaluation metrics are as follows:

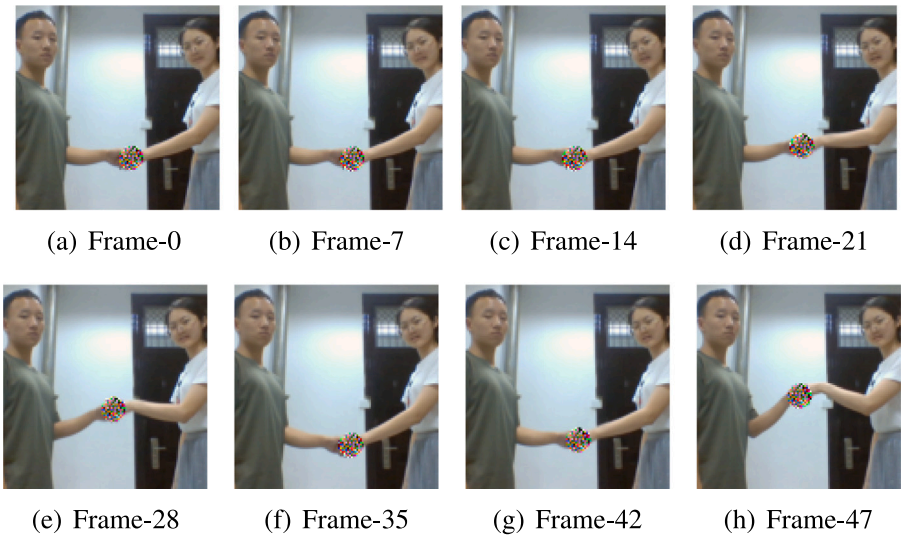


Fig. 14. Figure for results that generate in the digital world. And we sample each frame every seven frames of 48-frames input. That is 0,7,14,21,28,35,42,47. The original label is 'shake-hands', after putting the patch on the hands, the label is 'shoot-gun'.

Table 2
Table for comparing the result of different approaches. For the limitation of the length of the table, we simplified the method name. "Wei2018" is the method [17], "Zajac2019" is the method [30] and "Zhang2020" is the approach in [29].

Metrics	Method			
	Wei2018	Zajac2019	Zhang2020	Ours
Consistent value	No	No	No	Yes
Motional location	No	No	No	Yes
Able to physical attack	No	No	No	Yes

- Fooling Rate (FR): The percentage of adversarial videos that are successfully misclassified [22]. In the target attack, the misclassified label must be the target label. In an untarget attack, the misclassified label should be different from the original label. A larger value represents a better attack.
- Space (SP): The area of the patch, we use the radius of a circle as the measure of the area. A smaller value means smaller perturbations, and the patch will be more imperceptible.
- Iterations (IT): The average number of iterations for generating perturbations. A small value means a fast attack.
- Time (TI): The time of required to attack a video. A small value denotes a fast attack.

5.2. Adversarial perturbations in the digital world

Compared with other digital world AEs generation methods, this algorithm can generate digital world adversarial videos which is spatial motional and temporally consistent. As shown in Table 2, we give the comparison between our algorithm and other digital world video AEs generation algorithms. Our perturbations are motional, whose location changes with the hands. And the value of perturbations keeps consistent during the video playing.

The digital world AEs experiment tests the experimental results of adversarial videos without transformation of physical world, as shown in Table 3. In order to ensure the consistency of experimental conditions, we set the 'S = 0%' in the work [17] attack, and 'W = 4' in the [30]. Fig. 14 visualize the experimental results of the digital world AEs.

Our algorithm achieves an 100% attack success rate under the test settings. According to the evaluations, we make the following observations. (a) With the increase in SP, the success rate FR of the attack increases. We notice that the FR of single-video training under the 55 SP setting of the C3D model is 0.78, but it is 1.00 with iterative training. The iterative training strategy uses the 48 frames of each video to iteratively train the patch, while the single-video training process only uses the first 48 frames to train the patch. Therefore, the iterative training method can enhance the strength of adversarial noise and improve the success rate of the attack. (b) In the current settings used for our experiments, the best SPs of different models are different. Our experiments can provide an indication for setting different mask sizes. Moreover, no direct relation is observed between the size of the mask and the experimental performance. We generally believe that the larger the mask is, the better the performance. However, the experimental results show that this is not the case. When the mask size is too large, excessive noise is produced, so the algorithm's optimization process needs to "pull the noise back" to a smaller value, which leads to a reduction in efficiency. (c) The best SPs of single-video training and iterative training are also different. The reason for this finding is the same as that in '(a)'.

5.3. Adversarial perturbations in physical world

In this section, we test our algorithm in real physical settings. We test it on a real-time dynamic video recognition model, which can clearly demonstrate the model performance. The real-time recognition method is more suitable for applications involving actual scene. Fig. 15 shows the adversarial results obtained in the physical world.

To test the experimental results under different distances, we stick the patch on the hands and stand at different distances in front of the camera. We record 10 videos while keeping the other physical environment variables consistent. Then, we test the FR. The results are shown in Table 4. Specifically, the FR metric is the untarget attack accuracy.

As Table 4 shows, distance changes have significant impacts on adversarial attacks. Although the training process includes distance variation during the generation of adversarial videos, the distance still plays a vital role in the adversarial features of perturbations. In addition to the decrease in the percentage of perturbations relative to the whole space caused by the increase in distance, two other factors are present: the cameras fails to accurately capture the perturbations, other non-adversarial pixels increase. If the training process does not address different distances, the performance becomes more unsatisfactory. The details are shown in the ablation study.

Lastly, we show our adversarial videos can destroy the temporal relations between the frames. We generate the Class Activation Maps [42]

Table 3
The results of the digital adversarial attacks. The arrow in the table means the performance of the value of metrics. The ↓ means it has a better performance when the value of the metrics is lower. The ↑ means it has a better performance when the value of the metrics is higher. The reason is illustrated in Section Metrics.

Metrics	Method								
	C3D			I3D			LRCN		
SP(↓)	25	40	60	25	40	60	25	40	60
FR(↑)	0.40	1.00	1.00	0.30	1.00	1.00	0.45	1.00	1.00
IT(↓)	50.27	73.67	82.73	70.00	72.00	48.72	88.76	96.6	82.22
TI(↓)	184.73	235.1	263.7	95.59	101.59	50.59	154.28	172.6	88.60

Fig. 15. The first line contains the results of attacking the real-time dynamic recognition model in the physical world. The second line contains the frames of the original videos.

Table 4
The experimental results obtained under different distances and speeds.

Distance/Speed	Wave	Shake-hands	Distance/Speed	Wave	Shake-hands
77.5 cm/60′			77.5 cm/80′		
77.5 cm/80′			122 cm/80′		
77.5 cm/100′			152 cm/80′		
77.5 cm/120′			182 cm/80′		
FR	77.5	52.5	FR	35	37.5

of our adversarial videos and the original videos, which are as shown in Fig. 16. From that figure, we can see that in the original video frames, whose video's label is 'wave', the attention map is moving with the action of the palm. However, that maps of adversarial video frames are focus on the same location, which like the 'fog' of the cigarettes.

6. Ablation study

In our algorithm, we use iterative training to eliminate the influence of different inputting style. In order to show the effectiveness of iterative training, we design ablation study as shown in Table 5. As Table 5 shows, the attack success rate of AEs after iterative training is higher than that of only using single video training. For C3D model, iterative training uses every 48 frames in the video as the trained input, while single video training only uses the first 48 frames as the trained input. When the SP is 40, the attack success rate of AEs obtained by single video training is 0.78, while ti is 1.00 for iterative training. Meanwhile, because iterative training improves the attack ability of AEs, the SP of the required dynamic mask can be reduced to a certain extent.

To prove the importance of training the Λ_s and Λ_d of the transform function, we test the performance achieved when losing one of these two parts and when losing both parts in the physical adversarial attack. We select 10 videos from the sampled videos recorded at different distances and different speeds. We test the FRs yielded under the three settings. As shown, we label the 'ls-attack' as the setting of losing the Λ_s during training, the 'ld-attack' as that of losing the Λ_d during training, and 'lds-attack' as losing all the two parts of training. The performance is shown as Fig. 17. We set the SP of that experiment is 55. We can clearly see that the Λ_s training and the Λ_d training can improve the FR.

We also generate the perturbations by using the traditional mask generation method, which is shown in Fig. 18. The perturbations of this method can also result in model misclassifications in the digital world. However, they cannot work in the physical world if we only choose one frame, clip the noise and then stick the patch on the hand. Although this method does not work well in the current situation, we suppose that if technology lets the perturbations change with the video within the same frame rate and defends the destruction of adversarial features due to the different sampling rates, the FR achieved under the target setting will increase.

We also design comparative experiment attacking the model by sticking the Gaussian Noise, which has the same size in the same location as others. Fig. 19 shows it.

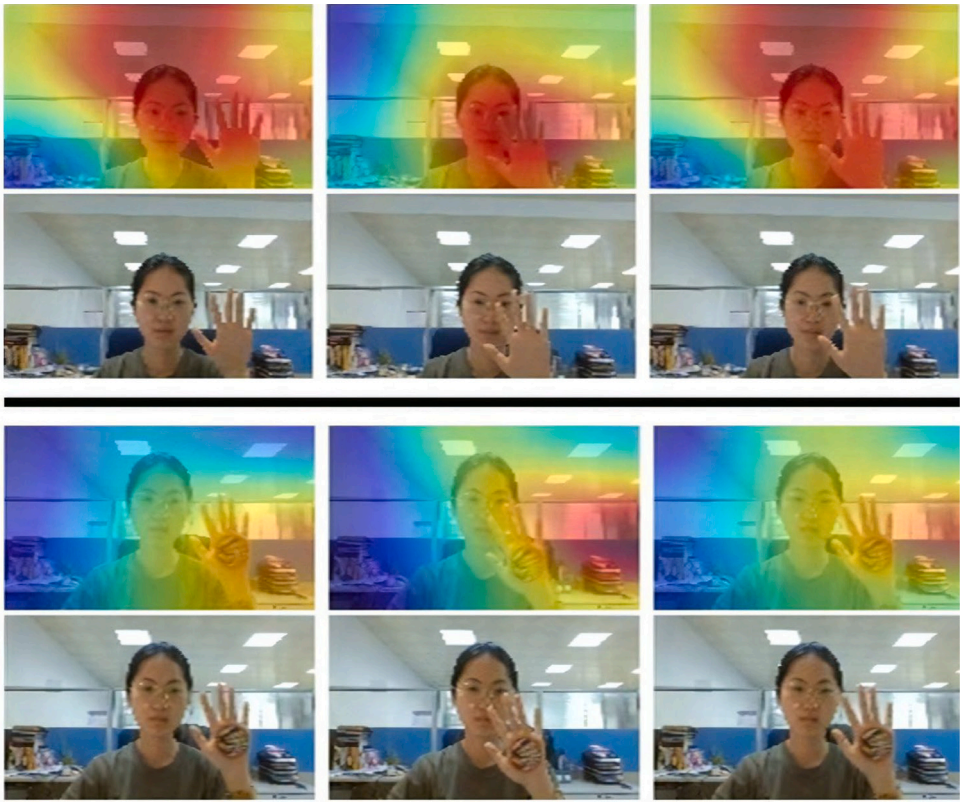


Fig. 16. The figure of class attention map of adversarial video frames and original video frames. The first and the third lines are the class attention map, and the second and the fourth lines are the original frames. The top six pictures are the clean video frames, whose label is 'wave'. The bottom six pictures are the adversarial video frames, whose label is 'smoke'. The adversarial video is generated by sticking a patch whose radius is 55 pixels.

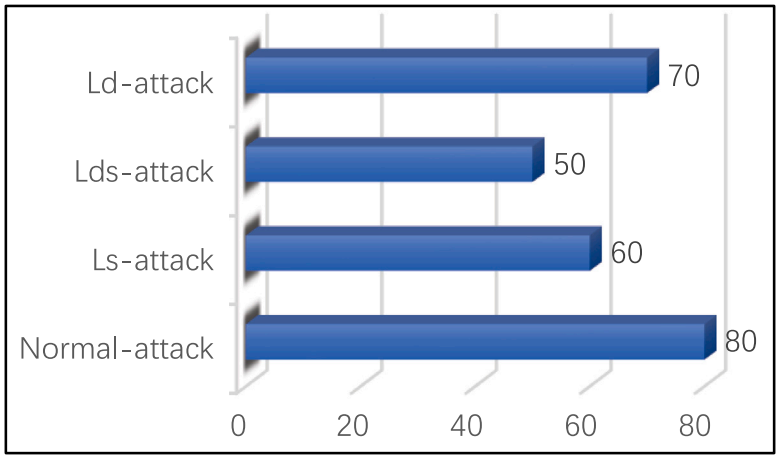
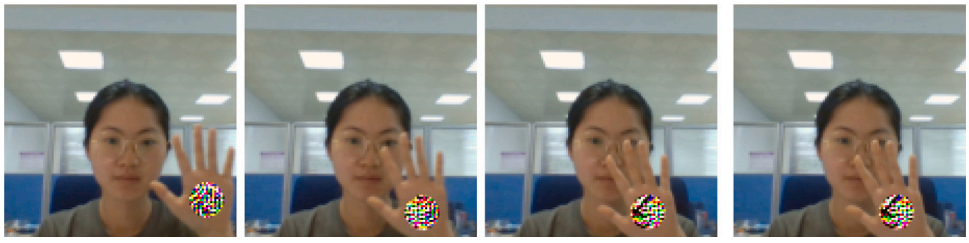


Fig. 17. The results of physical adversarial attacks under different settings.



(a) Frame-0 (b) Frame-7 (c) Frame-14 (d) Frame-21

Fig. 18. The figure of the results generated in the digital world by the traditional mask generation method.

Table 5
The results of the digital adversarial attacks. The iterative training process uses the 48 frames of each video to iteratively train the patch, while the single-video training process only uses the first 48 frames to train the patch. The arrow in the table means the performance of the value of metrics. The ↓ means it has a better performance when the value of the metrics is lower. The ↑ means it has a better performance when the value of the metrics is higher. The reason is illustrated in Section Metrics.

Model	Metrics							
	Single video training				Iterative training			
	SP(↓)	FR(↑)	IT(↓)	TI(↓)	SP(↓)	FR(↑)	IT(↓)	TI(↓)
C3D	40	0.62	100.04	2108.2	25	0.40	50.27	184.73
	45	0.78	92.00	1920.4	30	0.40	67.41	210.39
	50	0.85	81.25	1719.0	45	0.78	71.22	233.82
	55	1.00	59.72	772.7	50	**1.00**	**70.33**	**219.98**
	60	**1.00**	**53.00**	**720.01**	55	1.00	82.73	263.7
I3D	25	0.54	30.09	40.28	25	0.72	70.00	95.59
	35	0.54	32.94	49.29	35	0.78	63.33	60.30
	40	0.78	32.28	43.86	40	0.78	72.00	101.59
	55	**1.00**	**33.22**	**52.04**	55	1.00	48.83	44.03
	60	1.00	33.78	51.33	60	**1.00**	**48.72**	**50.59**
LRCN	25	0.35	40.87	60.21	25	0.35	88.76	154.28
	35	0.35	42.56	65.04	45	0.40	80.94	79.32
	45	0.54	42.44	75.45	45	0.54	86.61	110.89
	55	**1.00**	**42.28**	**64.49**	55	**1.00**	**83.44**	**80.99**
	60	1.00	43.56	80.28	60	1.00	82.22	88.60

(a) Frame-0 (b) Frame-7

(c) Frame-14 (d) Frame-21

Fig. 19. Figure for attacking the real-time human recognition model by adding Gaussian Noise on the palm.

7. Conclusion

Our work is the first study to generate physical adversarial patches attacking real-time dynamic video recognition models. We also propose to develop a motional but temporally consistent perturbations method. We utilize improved object tacking methods to generate a motional mask and use the center location of the mask to obtain temporally consistent noise. We perform extensive experiments to evaluate our method in different settings and demonstrate its effectiveness in the physical world. However, we only take two actions as examples to verify our framework and give the algorithm for the two actions. In future, we will extend our work to more actions and increase the generality of our algorithm.

CRediT authorship contribution statement

Zhenyu Du: Conceptualization, Methodology, Software, Investigation, Formal Analysis, Writing – original draft. **Xingxing Wei:** Data curation, Writing – original draft. **Weiming Zhang:** Resources, Supervision, Writing – review & editing. **Fangzheng Liu:** Visualization, Investigation. **Huanyu Bian:** Supervision. **Jiayang Liu:** Software, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

A.1. Analysis of Eq. (1)

In this section, we give the detailed derivation of Eq. (1). First we recalled the Shoelace Theorem as Theorem 1.

Theorem 1 (*Shoelace Theorem*). *Suppose the polygon P has vertices $(a_1, b_1), (a_2, b_2), \ldots (a_n, b_n)$, listed in clockwise order. Then the area (A) of P is*

$$A = \frac{1}{2}\left|\left(a_1 b_2 + a_2 b_3 + \cdots + a_n b_1\right) - \left(b_1 a_2 + b_2 a_3 + \cdots + b_n a_1\right)\right| \tag{A.1}$$

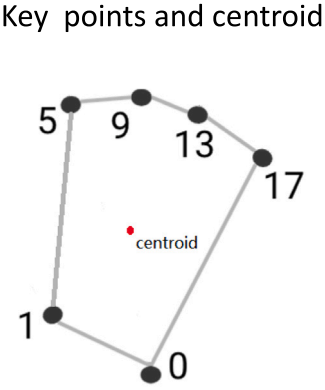


Fig. A.20. Figure for polygon and centroid of Eq. (1).

The centroid of a non-self-intersecting closed polygon defined by n vertices $(a_1, b_1), (a_2, b_2), \ldots (a_n, b_n)$ is the point (C_x, C_y) is

$$\begin{aligned} C_x &= \frac{1}{6A}\sum_{i=0}^{n-1}(a_i + a_{i+1})(a_i b_{i+1} - a_{i+1} b_i), \\ C_y &= \frac{1}{6A}\sum_{i=0}^{n-1}(b_i + b_{i+1})(a_i b_{i+1} - a_{i+1} b_i), \end{aligned} \tag{A.2}$$

According to Eq. (A.2) and the definition of key points in Section 4.1, we have the following derivation. Give n key points $(\xi_0, \psi_0), (\xi_1, \psi_1), \ldots (\xi_{n-1}, \psi_{n-1})$, they can get the polygon shown in the middle picture of Fig. 8. For easy understanding, we redisplay it in Fig. A.20.

Putting Eqs. (A.1) to Eq. (A.2), we have the following proof of Eq. (1).

Proof. The centroid (Ξ, Ψ) is

$$\begin{aligned} \Xi &= \frac{1}{6A}\sum_{i=0}^{n-1}(\xi_i + \xi_{i+1})(\xi_i\psi_{i+1} - \xi_{i+1}\psi_i) \\ &= \frac{1}{6(\frac{1}{2}\left|\left(\sum_{i=0}^{n-1}(\xi_i\psi_{i+1} - \psi_i\xi_{i+1})\right)\right|)}\sum_{i=0}^{n-1}(\xi_i + \xi_{i+1})(\xi_i\psi_{i+1} - \psi_{i+1}\psi_i) \\ &= \frac{\sum_{i=0}^{n-1}(\xi_i\psi_{i+1} - \xi_{i+1}\psi_i)(\xi_i + \xi_{i+1})}{3\sum_{i=0}^{n-1}(\xi_i\psi_{i+1} - \xi_{i+1}\psi_i)} \end{aligned} \tag{A.3}$$

$$\begin{aligned} \Psi &= \frac{1}{6A}\sum_{i=0}^{n-1}(\psi_i + \psi_{i+1})(\xi_i\psi_{i+1} - \xi_{i+1}\psi_i) \\ &= \frac{1}{6(\frac{1}{2}\left|\left(\sum_{i=0}^{n-1}(\xi_i\psi_{i+1} - \psi_i\xi_{i+1})\right)\right|)}\sum_{i=0}^{n-1}(\psi_i + \psi_{i+1})(\xi_i\psi_{i+1} - \psi_{i+1}\psi_i) \\ &= \frac{\sum_{i=0}^{n-1}(\xi_i\psi_{i+1} - \xi_{i+1}\psi_i)(\psi_i + \psi_{i+1})}{3\sum_{i=0}^{n-1}(\xi_i\psi_{i+1} - \xi_{i+1}\psi_i)} \end{aligned} \tag{A.4}$$

Appendix B. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jisa.2022.103278.

References

[1] Eykholt K, Evtimov I, Fernandes E, Li B, Rahmati A, Xiao C, et al. Robust physical-world attacks on deep learning models. 2017, arXiv:1707.08945.

[2] Liu A, Wang J, Liu X, Cao B, Zhang C, Yu H. Bias-based universal adversarial patch attack for automatic check-out. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). LNCS, vol. 12358, 2020, p. 395–410. http://dx.doi.org/10.1007/978-3-030-58601-0_24, arXiv:2005.09257.

[3] Bontrager P, Roy A, Togelius J, Memon N, Ross A. DeepMasterPrints: GEnerating masterprints for dictionary attacks via latent variable evolution. In: 2018 IEEE 9th international conference on biometrics theory, applications and systems, BTAS 2018. 2018, http://dx.doi.org/10.1109/BTAS.2018.8698539, arXiv:1705.07386.

[4] Fawzi A, Fawzi O, Frossard P. Analysis of classifiers' robustness to adversarial perturbations. Mach Learn 2018;107(2).

[5] Moosavi-Dezfooli SM, Fawzi A, Frossard P. DeepFool: A simple and accurate method to fool deep neural networks. In: CVPR. 2016, p. 2574–82. http://dx.doi.org/10.1109/CVPR.2016.282, arXiv:arXiv:1511.04599v3.

[6] Zhang B, Tondi B, Barni M. Adversarial examples for replay attacks against CNN-based face recognition with anti-spoofing capability. Comput Vis Image Underst 2020;197–198:102988.

[7] Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. In: 3rd International conference on learning representations, ICLR 2015 - conference track proceedings. 2015, p. 1–11, arXiv:1412.6572.

[8] Chung HY, Chung YL, Tsai WF. An efficient hand gesture recognition system based on deep CNN. In: Proceedings of the IEEE international conference on industrial technology, 2019-February. 2019, p. 853–8. http://dx.doi.org/10.1109/ICIT.2019.8755038.

[9] Li G, Tang H, Sun Y, Kong J, Jiang G, Jiang D, et al. Hand gesture recognition based on convolution neural network. Cluster Comput 2019;22:2719–29. http://dx.doi.org/10.1007/s10586-017-1435-x.

[10] Wu XY. A hand gesture recognition algorithm based on DC-CNN. Multimedia Tools Appl 2020;79(13–14):9193–205. http://dx.doi.org/10.1007/s11042-019-7193-4.

[11] Nguyen XS, Brun L, Lezoray O, Bougleux S. A neural network based on SPD manifold learning for skeleton-based hand gesture recognition. 2019, arXiv.

[12] Bao P, Maqueda AI, Del-Blanco CR, Garciá N. Tiny hand gesture recognition without localization via a deep convolutional network. IEEE Trans Consum Electron 2017;63(3):251–7. http://dx.doi.org/10.1109/TCE.2017.014971.

[13] Eykholt K, Evtimov I, Fernandes E, Li B, Arbor A, Rahmati A, et al. Robust physical-world attacks on deep learning visual classification. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition. 2018, p. 1625–34. http://dx.doi.org/10.1109/CVPR.2018.00175, arXiv:arXiv:1707.08945v5.

[14] Lee M, Kolter Z. On physical adversarial patches for object detection. In: ICML 2019 workshop on security and privacy of machine learning. 2019, arXiv:1906.11897.

[15] Thys S, Van Ranst W, Goedemé T. Fooling automated surveillance cameras: adversarial patches to attack person detection. In: CVPRW: workshop on the bright and dark sides of computer vision: challenges and opportunities for privacy and security. 2019, http://dx.doi.org/10.1109/CVPRW.2019.00012, arXiv:1904.08653.

[16] Li S, Neupane A, Paul S, Song C, Krishnamurthy SV, Chowdhury AKR, et al. Adversarial perturbations against real-time video classification systems. 2018, arXiv preprint arXiv:1807.00458, http://dx.doi.org/10.14722/ndss.2019.23202.

[17] Wei X, Zhu J, Yuan S, Su H. Sparse adversarial perturbations for videos. In: AAAI, Vol. 33. 2019, p. 8973–80. http://dx.doi.org/10.1609/aaai.v33i01.33018973, arXiv:1803.02536.
[18] Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, et al. Intriguing properties of neural networks. 2013, arXiv:1312.6199.
[19] Papernot N, McDaniel P, Swami A, Harang R. Crafting adversarial input sequences for recurrent neural networks. In: IEEE military communications conference. 2016, p. 49–54. http://dx.doi.org/10.1109/MILCOM.2016.7795300, arXiv:1604.08275.
[20] Kurakin A, Goodfellow I, Bengio S. Adversarial examples in the physical world. 2016, arXiv e-printsarXiv:1607.02533.
[21] Carlini N, Wagner D. Towards evaluating the robustness of neural networks. In: IEEE symposium on security and privacy. 2017, p. 39–57, arXiv:1608.04644, http://dx.doi.org/10.1109/SP.2017.49.
[22] Moosavi-Dezfooli SM, Fawzi A, Fawzi O, Frossard P. Universal adversarial perturbations. In: CVPR. 2017, p. 86–94. http://dx.doi.org/10.1109/CVPR.2017.17, arXiv:arXiv:1610.08401v2.
[23] Chen PY, Zhang H, Sharma Y, Yi J, Hsieh CJ. ZOO: ZEroth order optimization based black-box atacks to deep neural networks without training substitute models. In: AISec. 2017, p. 15–26. http://dx.doi.org/10.1145/3128572.3140448, arXiv:arXiv:1708.03999v2.
[24] Tu C, Ting P, Chen P, Liu S, Zhang H, Yi J, et al. AutoZOOM: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. 2018, CoRR arXiv:1805.11770.
[25] Brendel W, Rauber J, Bethge M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In: ICLR. 2018, p. 1–12, arXiv:1712.04248.
[26] Cheng M, Zhang H, Hsieh CJ, Le T, Chen PY, Yi J. Query-efficient hard-label black-box attack: An optimization-based approach. In: ICLR. 2019, p. 1–12, arXiv:1807.04457.
[27] Jiang L, Ma X, Chen S, Bailey J, Jiang YG. Black-box adversarial attacks on video recognition models. In: ACMMM. 2019, p. 864–72. http://dx.doi.org/10.1145/3343031.3351088, arXiv:1911.09449.
[28] Wei Z, Chen J, Wei X, Jiang L, Chua T-S, Zhou F, et al. Heuristic black-box adversarial attacks on video recognition models. In: IJCAI. 2019, p. 864–72. http://dx.doi.org/10.1145/3343031.3351088, arXiv:1911.09449.
[29] Zhang H, Zhu L, Zhu Y, Yang Y. Motion-excited sampler: Video adversarial attack with sparked prior. In: European conference on computer vision. Vol. 1. 2020, arXiv:2003.07637.
[30] Zajac M, Zołna K, Rostamzadeh N, Pinheiro PO. Adversarial framing for image and video ClassificatiPrevious work proves the effectiveness of *sparse attack*, which add the perturbations on the sparse frames.on. In: Proceedings of the AAAI conference on artificial intelligence, Vol. 33. 2019, p. 10077–8. http://dx.doi.org/10.1609/aaai.v33i01.330110077, [17] arXiv:1812.04599.
[31] Sharif M, Bhagavatula S, Bauer L, Reiter MK. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the ACM conference on computer and communications security, 24-28-Octo. 2016, p. 1528–40. http://dx.doi.org/10.1145/2976749.2978392.
[32] Komkov S, Petiushko A. AdvHat: Real-world adversarial attack on ArcFace face ID system. 2019, arXiv preprint arXiv:1908.08705.
[33] Xu K, Zhang G, Liu S, Fan Q, Sun M, Chen H, et al. Adversarial T-shirt! evading person detectors in a physical world. In: European conference on computer vision. 2020, arXiv:1910.11099.
[34] Xu K, Zhang G, Liu S, Fan Q, Sun M, Chen H, et al. Adversarial T-shirt! evading person detectors in a physical world. 2019, arXiv e-prints arXiv:1910.11099.
[35] Nassi B, Nassi D, Ben-netanel R, Mirsky Y, Drokin O, Elovici Y. Phantom of the ADAS : Phantom attacks on driver-assistance systems. In: Cryptology eprint archive. 2020, p. 1–17, URL https://eprint.iacr.org/2020/085.
[36] Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T. HMDB: A Large video database for human motion recognition H.. In: HLRS. 2011, p. 311–24. http://dx.doi.org/10.1007/978-3-642-33374-3.
[37] Zhang F, Bazarevsky V, Vakunov A, Tkachenka A, Sung G, Chang C-L, et al. MediaPipe Hands: On-device real-time hand tracking. 2020, arXiv preprint arXiv:2006.10214.
[38] JuliaPoo. Multihand-tracking. 2020, GitHub Repository, https://github.com/JuliaPoo/MultiHand-Tracking.
[39] Carreira J, Zisserman A. Quo vadis, action recognition? A new model and the kinetics dataset. In: Proceedings - 30th IEEE conference on computer vision and pattern recognition, CVPR 2017, 2017-January. 2017, p. 4724–33. http://dx.doi.org/10.1109/CVPR.2017.502, arXiv:1705.07750.
[40] Tran D, Bourdev L, Fergus R, Torresani L, Paluri M. Learning spatiotemporal features with 3D convolutional networks. In: ICCV, 2015 inter. 2015, p. 4489–97. http://dx.doi.org/10.1109/ICCV.2015.510, arXiv:1412.0767.
[41] Donahue J, Hendricks LA, Rohrbach M, Venugopalan S, Guadarrama S, Saenko K, et al. Long-term recurrent convolutional networks for visual recognition and description. IEEE Trans Pattern Anal Mach Intell 2017;39(4):677–91. http://dx.doi.org/10.1109/TPAMI.2016.2599174, arXiv:1411.4389.
[42] Chattopadhay A, Sarkar A, Howlader P, Balasubramanian VN. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In: Proceedings - 2018 IEEE winter conference on applications of computer vision, WACV 2018, 2018-Janua. 2018, p. 839–47. http://dx.doi.org/10.1109/WACV.2018.00097, arXiv:arXiv:1710.11063v3.