Information Sciences 546 (2021) 596–607

Contents lists available at ScienceDirect

Information Sciences

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

A discrete cosine transform-based query efficient attack on black-box object detectors

Xiaohui Kuang [a,c], Xianfeng Gao [a,b], Lianfang Wang [a], Gang Zhao [c], Lishan Ke [d], Quanxin Zhang [a,*]

[a] School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
[b] Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou, 510006, China
[c] National Key Laboratory of Science and Technology on Information System Security, Beijing, China
[d] School of Computer Science, Guangzhou University, Guangzhou, China

ARTICLE INFO

Article history:
Received 21 November 2019
Received in revised form 28 February 2020
Accepted 20 May 2020
Available online 9 September 2020

Keywords:
Object detection
Boundary attack
Discrete cosine transform (DCT)
Black-box
Adversarial example

ABSTRACT

Deep learning models are being widely used in almost every field of computing and information processing. The advantages offered by these models are unparalleled, however, similar to any other computing discipline, they are also vulnerable to security threats. A compromised deep neural network can significantly impact its robustness and accuracy. In this work, we present a novel targeted attack method against state-of-the-art object detection models YOLO v3 and AWS Rekognition in a black-box environment. We present an improved attack method using Discrete Cosine Transform based on boundary attack plus plus mechanism, and apply it on attacking object detectors offline and online. By querying the victim detection models along with transforming the images from the spatial domain into the frequency domain, we ensure that any specified object in an image can be successfully recognized as any other desired class by YOLO v3 and AWS Rekognition. The results prove that our method has significant boosting effects on boundary attacks in offline and online object detection systems.

1. Introduction

The use of deep learning in modern applications has increased significantly in the past few years. The rapid development of artificial intelligence applications, such as computer vision, language recognition, secure communication, traffic safety, natural language processing, autonomous vehicles, etc., has further increased the use of deep learning and neural networks as a core supporting technology [1–5]. The scope of its use is not limited to these, rather many other research works have used it in other application areas as well [6–9]. Given the tremendous benefits offered by deep learning, it is also prone to several security problems. Work by Szegedy et al. [10] was the first to describe the different vulnerabilities of the deep neural networks. The simplest and easiest method of attacking a deep learning network is through *adversarial example*. This attack can cheat the deep learning classifier by disturbing very little in the image, while the changes are not perceptible to the human eye. Hence, the classifier outputs a wrong label for the adversarial example with a high confidence level. The same adversarial phenomenon can occur in object detection [2]. For example, if the attacker changes the traffic signs to an adversarial image, the unmanned vehicle may ignore or incorrectly detect the traffic information, which may lead to serious consequences.

[*] Corresponding author.
E-mail address: zhangqx@bit.edu.cn (Q. Zhang).

https://doi.org/10.1016/j.ins.2020.05.089

Image classification and object detection are two major applications of deep learning technologies. The attack methods in this regard can be classified as white-box or black-box attacks [11–14]. For the white-box attacks, the attacker can analyze the details of the victim model, hence, the attackers can find different opportunities to implement their attack. The white-box attack methods also form the fundamental principle of the black-box attacks. On the other hand in a black-box method, the attackers cannot obtain the details of the victim model such as the infrastructure, parameters, defense techniques, etc. The only returned information is the output label (in some cases along with the confidence level). Thus, it is significantly difficult to attack the black-box model, which is why most of the applied real deep neural networks are black-box.

Several works on white-box methods are available in the literature, hence we here we only describe the most significant contributions in this regard. Szegedy et al. [10] proposed an L-BFGS attack based on the simple bound constraint. The Fast Gradient Sign Method (FGSM) was initially designed by Goodfellow et al. [15], while Kurakin et al. [16] improved it by proposing an Iterative Fast Gradient Sign Method (I-FGSM) attack. Work of Dong et al. [17] significantly improved the generation of adversarial example by Momentum Iterative Fast Gradient Sign Method (MI-FGSM), while the Jacobian-based Saliency Map Attack (JSMA) was proposed by Papernot et al. for targeted attacks [18]. Carlini and Wagner [19] presented a C & W method (Carlini and Wagner Attack) which has a very high success rate.

The black-box model attacks are less researched due to difficulty in execution as discussed earlier. The work by Papernot et al. in [20] presented a substitution model attack method by utilizing the transferability of the adversarial example. It first generates the adversarial examples using a white-box technique on the substitution model and then applies the attack on the black-box model. Su et al. [21] proposed the One Pixel attack using an adversarial example, and claims to have 98.7% average confidence by only modifying one pixel in the test image. The Boundary Attack (BA) was proposed by Brendel et al. [22] for the black-box attacks, and only depends on the outputs (such as the classification label of top-1) to successfully execute the attack.

It is important to note that as the use of deep learning techniques in different application domains has increased [23–27], hence the attacks on image classifiers have also become prominent. The research on adversarial examples can improve the security and safety of deep learning systems. Based on this motivation, in this work, we present a boundary attack in the frequency domain based on Discrete Cosine Transform (DCT) towards black-box object detectors. This attack generates robust object adversarial examples, which can then be used for attacking the system. The major contributions and highlights of this work are as follows.

- It is difficult to attack the black-box object detectors since the feedback information is very limited. Most of the black-box attacks are based on Gradient Descent, Boundary Attack, or other alternative models. To the best of our knowledge, this is the first work that proposes targeted black-box attacks on object detection systems based on DCT.
- An object detector can recognize the position, size, and category of the objects in the image, while it is difficult to disturb the detector for designated objects since the attacker is unaware of the details of the detectors. Most of the methods implement an indiscriminate attack for all objects in the image. This is the first work, which can attack a single specific object rather than the total image on object detectors.
- The query times are the key indicator for the black-box attacks, which leads to attack concealment and practical feasibility. In the proposed approach, we successfully increase the query efficiency of Boundary Attack via DCT by decreasing the query times.

The remainder of the paper is organized into six sections. Section 2 briefly introduces the background of essential technologies for generating adversarial examples. Following this, Section 3 explains the principles of Discrete Cosine Transform and its advantages on image processing. It also describes the working of the proposed object detector. The image preprocessing and output processing are described in Section 4 for applying the DCT to image modification. The improved boundary attack based is elaborated in Section 5. Section 6 presents the experiments and the efficiency of the proposed scheme. Finally, the conclusion of this work is given in Section 7.

2. Background

An imperceptible perturbation is usually added to a natural image by attackers to change a model's prediction while studying adversarial examples in image classification. One of the reasons why there are adversarial examples is that the training set cannot cover all classification or detection probabilities. Hence, most of the neural models have inherent shortcuts.

2.1. White-box attacks

The attacker knows everything about the victim model, including its parameters, infrastructures, training process, and training dataset. Hence, the generation of adversarial example can be treated as an optimization problem, given as:

$$|x^{adv} - x|_2 \rightarrow minimum, \; s.t \;\; f\left(x^{adv}\right) = y_{target}, \;\; x^{adv} \in [0,1]^m \tag{1}$$

There are several white-box attack methods that solve this problem, such as L-BFGS. however, most of these methods are time-consuming. Goodfellow et al. [15] proposed the FGSM for verifying the adversarial examples that can be generated only by linear approximation of the model. FGSM can generate adversarial examples quickly and is suitable for non-targeted attacks. The disturbances in the image are added by back-propagation for modifying the inputs. It can be expressed as:

$$\eta = \alpha sign\left(\nabla_x J(x_t^*, y)\right) \tag{2}$$

L-BFGS can attain a high attack success rate with high computation cost, while FGSM can generate adversarial examples faster with a low attack success rate. Basic Iterative Method (BIM) [28] can be considered as a middle ground between computation cost and success rate. BIM is an iterative method based on FGSM. Using it, a temporary adversarial example $\left(x_{(t+1)}^* = x_t^* + \alpha g\right)$ is obtained after each iteration, while x_T is the final adversarial example after T iterations.

Carlini and Wagner [19] presents a new loss function based on L-BFGS. It can generate adversarial examples with low $L_2, \ L_\infty, \ L_0$. The adversarial examples with high adversary are the examples with low L_2, and its loss function is given as:

$$||x^{adv} - x||_p + cmax\left(\max_{i \neq Y} f\left(x^{adv}\right)_i - f\left(x^{adv}\right)_Y, -\kappa\right) \rightarrow minimum \tag{3}$$

2.2. Black-box attacks

The mechanism of black-box attacks can be divided into two branches. The first one transfers the adversarial example generation to a convex optimization problem, such as BA. While the second one utilizes the white-box attack techniques for the generation of adversarial examples and then optimize them via substitution models.

BA utilizes the reject sampling algorithm combined with recommendation distribution and trust-region heuristic for adjusting the step size dynamically. It always starts from an adversarial example with a high disturbance level and reduces the disturbances while keeping antagonism. BA is also an important principle of other black-box attacks.

Chen et al. [29] promoted the C & W attack, and proposed the ZOO algorithm. It can be regarded as a zero-order random coordinate descent method, which randomly selects a coordinate pixel and then estimates the gradient under the current coordinate. Decision-based attacks [22,30] are applicable in all scenarios as they utilize only the discrete classification decision models.

3. The principle of attack method

3.1. Discrete cosine transform

Discrete Cosine Transform (DCT) [31] is widely used mechanism in image compression [32]. The DCT coefficient energy is mainly concentrated in the upper left corner of the image, and most of the other coefficients are close to zero after implementing the DCT on the source image. The distance between adversarial example and source image can be converged faster in boundary attack based on DCT, which is proven through the experiments in this work. The main DCT principles and their application in our attack method are introduced in this section.

a) *One-dimensional DCT:* There are a total of eight forms in the one-dimensional DCT, and out of these, the second one is most commonly used due to its simple operation and wide adaptability. The main form used in the experiment presented in this work is given as follows.

$$F(u) = c(u)\sum_{i=0}^{N-1} f(i)cos\left[\frac{(i+0.5)\pi}{N}u\right] \tag{4}$$

$$c(u) = \begin{cases} \sqrt{\frac{1}{N}}, u = 0 \\ \sqrt{\frac{2}{N}}, u \neq 0 \end{cases} \tag{5}$$

where, $f(i)$ is the source signal, $F(u)$ is the coefficient after transformation, and N is the point number of source signal. The $c(u)$ can be treated as a compensation coefficient to transform DCT into orthogonal matrix.

b) *Two-dimensional DCT:* This makes another transformation based on the one-dimensional DCT. Its process is expressed as follows.

$$F(u, v) = c(u)c(v)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} f(i,j)cos\left[\frac{(i+0.5)\pi}{N}u\right]cos\left[\frac{(j+0.5)\pi}{N}v\right] \tag{6}$$

$$c(u) = \begin{cases} \sqrt{\frac{1}{N}}, u = 0 \\ \sqrt{\frac{2}{N}}, u \neq 0 \end{cases} \tag{7}$$

There are two basic features that make DCT suitable for the proposed attacking method. These are Reversibility and Distance preservation, which enable the proposed algorithms to be more efficient as proven in later sections.

a) *Reversibility:* The image can be restored from the frequency domain to the spatial domain during the process with this characteristic. The transformation and inverse transformation are similar, and we will introduce its core algorithm in the next section.

b) *Distance preservation:* The main principle of boundary attack is pushing the adversarial example close to source image while keeping antagonism. L_2 is chosen as the standard of distance measurement. DCT can preserve L_2 in that process as follows.

$$dist(a, b) = dist(DCT(a),\ DCT(b)) \tag{8}$$

This excellent property can push the adversarial example close to the source image in the frequency domain while keeping initial distance in the spatial domain. We can reduce the complexity of the programs for the DCT algorithm by simple matrix because of the symmetry of DCT.

$F = AFA^T$

$$A(i,j) = c(i)cos\left[\frac{j + 0.5\pi}{N} i\right] \tag{9}$$

where, $f = A^T FA$ is implemented if inverse transformation is desired.

3.2. Object detection

The object detection has many similarities with the classification system since it is adopted and upgraded from the machine learning classification model. The image is an input to the object detection system in a black-box environment and a detective result is an output. However, the object detection system has different characteristics as compared to classification. For example, classification outputs a label that points out which class the image belongs to, or outputs a confidence vector that has corresponding probabilities of the input image. While the object detection must detect not only one object but as many as possible independent objects in that image, and calculate the accurate positions of each object. Normally there are three messages returned from object detection in the black-box environment, i.e., labels, boxes, and scores, where they correspond to class, position, and confidence of the classification model.

4. Image preprocessing and output processing

The attack methods in image classification cannot be directly applied to object detection due to their inherent differences. Hence, image processing and output processing must be specifically implemented to realize targeted and non-targeted attacks in object detection. In this section, we give details regarding these processes.

4.1. Object selection

The first step in targeting a specific object is to appropriately position the image which contains it. Attacks in classification only require the image to be selected, while object attacks must also specify the object number within the selected image. The attacker inputs the image into the detector and then positions the object attacked after getting the output.

4.2. Preprocessing for source image

The basic principle of evolution or boundary attack (in classification) in the black-box environment, is to start from the target image and reduce the distance between the adversarial example and the source image. This is done by using pixels of the source image in high dimensional space to get better visual effects while keeping the adversary label (target label).

Contrary to this principle, the challenge in attacks for object detection is that it cannot start from the targeted image directly since the position of the targeted object in its image is usually different from the position of source object in the source image. For example, Fig. 1 shows two images: cars on the left side and a dog on the right side. If the attack objective is to misclassify a car in the lower right corner of the left image with a dog, where the dog is selected from the right image as the target object, then it can be observed that the position of the two objects in their respective images is quite different. Hence, one cannot get the correct attack result if the dog in the right image is selected as the starting point of the attack.

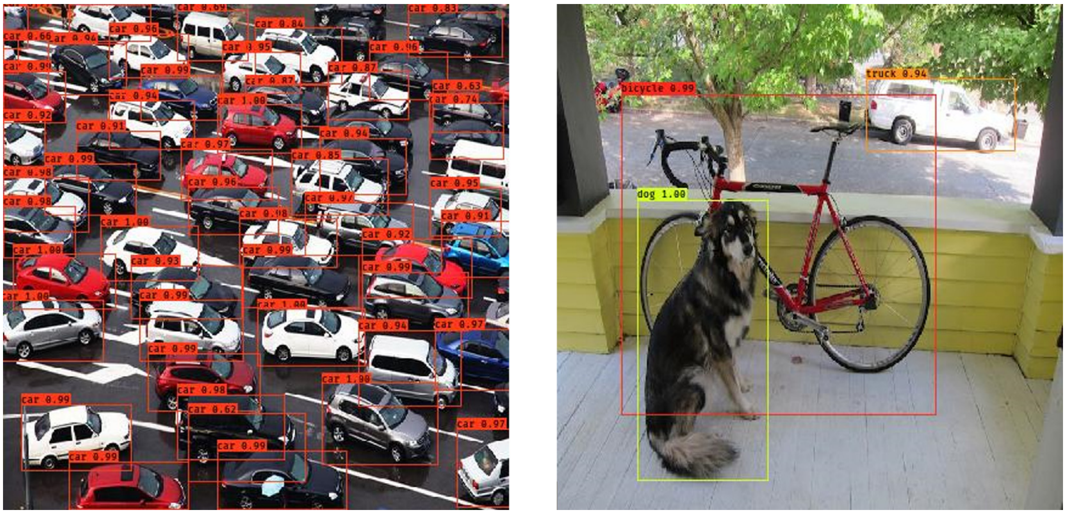


Fig. 1. The misaligned position of the source object (dog) and the target object (car).

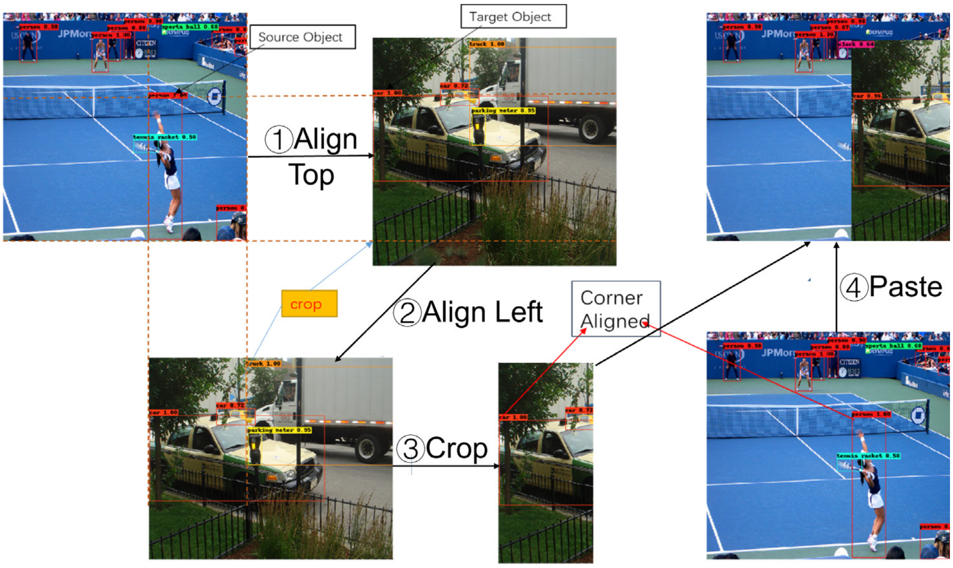


Fig. 2. The schematic diagram of corner alignment.

In this work, we deal with this problem using the "*corner alignment*" process. The schematic diagram of this process is given in Fig. 2 below, and the specific operations of corner alignment are:

1) Align the square box of the attacked source object with the square box of the targeted object on the upper left corner in the targeted image, and cut off the exceeded parts of the targeted object compared with the source image.
2) Cover the reminder targeted object on the source object in the source image while keeping the upper left corner of the two objects aligned.

By using this mechanism, we ensure that the targeted object is contained within the source image, and the source object and targeted object are aligned at the center point (roughly) after processing the source image. This type of covered image is defined as "*initial image*" and is the foundation of transformation. Following this, the distance between the initial image and source image will be reduced by boundary attack in the frequency domain to improve the visual effect.

4.3. Frame definition

There are two uncertainties during the attack. i) The detector uses Non-maximum Suppression (NMS) to search the objects in the image and outputs the results in the frames with the highest confidence. However, these results may be disturbed by tiny interference and it is difficult to keep the frame position completely the same as with the source image. ii) There may be multiple objects in an image, and the attacked object is not the original attacking target because of the change of the frame position. Hence, an effective way should be utilized to determine whether the attacked object is the original target during the attack process.

In this work, we use the center point to define whether the object in the frame is still the original attacked target in the experiment. We measure the coordinate of the center point of source object in the source image, and an effective adversarial example can be confirmed if the center point is in the frame of the targeted object during the preprocessing or attack.

5. Boundary attack in frequency domain

After completion of the preprocessing of the image, the DCT-based boundary attack in the frequency domain can be executed. In this section, we first describe the Boundary attack (BA) and its significantly improved version called *Boundary Attack Plus Plus* (BAPP). Following this, we present the novel DCT-based attack method.

5.1. BA and BAPP

Boundary attack is an effective decision-based attack method and has good adversarial result in most of the black-box environment. But it should be adjusted and improved according to the specific attack environment. For example, the black-box query times should be reduced significantly in a realistic attack environment since the classic BA needs thousands (and in some cases more than that) of query times which makes it inefficient and unpractical. The query process of the BA is shown in Fig. 3. Assume that the gray star is the adversary example starting image while the black star depicts the original image, as shown on the left side of the figure. The distance between the starting image and the original image will be reduced step by step while the gray star keeps its adversary. In the center part of the figure, the gray star will move in a random direction while keeping the distance between gray and black stars the same if it falls into local optimization and then projects towards the original image. On the right side of the figure, the gray star will adjust its step direction according to the adversary to avoid falling into the gray area. The iterative process of BA can be divided into two steps, given as following.

1) Project the adversarial example (targeted object) to the source image to make it closer to the classifier boundary. Iteration can stop if the L_2 distance between modified adversarial example and source image meets the threshold for the attack, else go to step (2).
2) Change the iteration direction far from boundary while keeping the adversarial characteristics, and return to step (1).

Most of the query time is consumed in searching for a suitable direction and step length. Chen et al. [33] proposed an improved attack method for BA, known as Boundary Attack Plus Plus (BAPP), which was renamed as HopSkipJumpAttack later. In this method, the adversary example is pushed by a decision function gradient direction as the most adversarial direction. The most suitable direction on the boundary is determined by several iterations of queries for fitting the gradient combined by the Zeroth Order Optimization (ZOO) technique. The difference is that the noise estimation is applied in BAPP, which is not done with BA. The decision function can be given as:

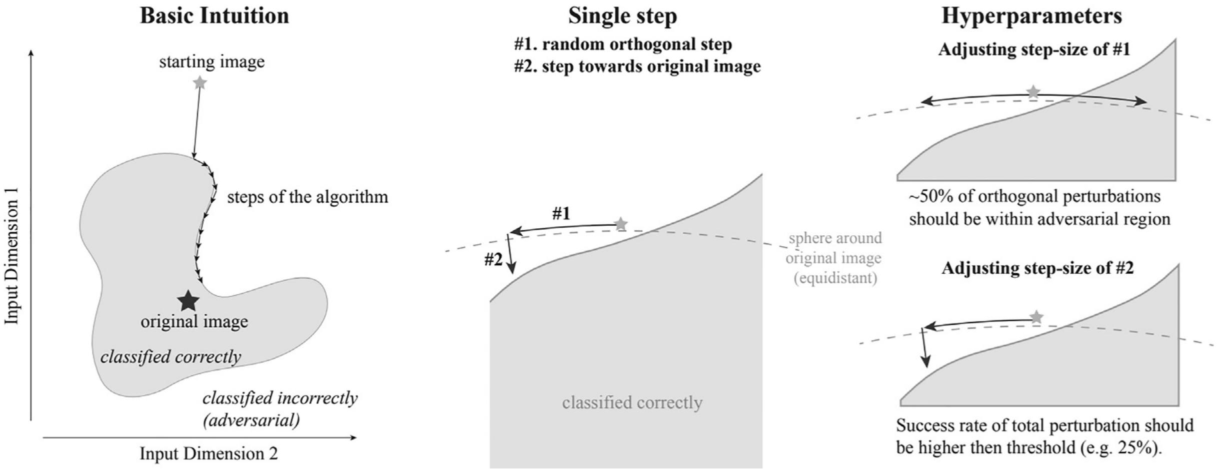

Basic Intuition
starting image
steps of the algorithm
original image
classified correctly
classified incorrectly
(adversarial)
Input Dimension 1
Input Dimension 2
Single step
#1. random orthogonal step
#2. step towards original image
#1
#2
sphere around
original image
(equidistant)
classified correctly
Hyperparameters
Adjusting step-size of #1
~50% of orthogonal perturbations
should be within adversarial region
Adjusting step-size of #2
Success rate of total perturbation should
be higher then threshold (e.g. 25%).

Fig. 3. A decision-based boundary attack to optimize the query process [22].

$$\widetilde{\nabla S}(x_t,\delta) := \frac{1}{B}\sum_{b=1}^{B}\phi_x * (x_t + \delta u_b)u_b, \tag{10}$$

where, ∇ is derivation operator, S is decision function, B is the query times for estimating the Gradient, ϕ is a decision function with binary result: 1 for adversary and 0 for non-adversary, u_i is orthogonal disturbance direction determined by estimated gradient, and δ is the estimated step length. The key of BAPP is in avoiding the local optimization and estimating the step length in step (2). This process is shown in Fig. 4. BAPP method significantly improves the boundary attack technique, and the proof can be found in [33].

5.2. Boundary attack based on DCT

Here, we present the novel attack method towards object detection according to the features of the object detector under BA infrastructure. The overall process can easily be understood in two parts..

First, we transform the BA attack from the spatial domain to the frequency domain according to the DCT distance preserving property during the adversarial example generation. In the proposed method, the image is transformed by DCT after its initialization, and then the BAPP is applied in order to reduce the noise. At the end of this mechanism, the final adversary example after inverse transformation by DCT under the noise limit can be obtained.

Second, the high frequency and low frequency of an image can be separated by DCT, where the low frequency is concentrated on the upper left corner of the image. The low frequency is most important for eyes or detectors, and the high frequency is usually ignored. Hence, significant attention to the low frequency of an image is given, and it is allocated more weight. Furthermore, as we can only modify the low frequency and ignore the high frequency, thus the low frequency of the image is modified by estimating only the gradient of low frequency. Finally, we substitute the binary decision function with Φ which ranges in [0,1] for improving the accuracy of the estimated gradient. The function for estimated gradient is given as:

$$\widetilde{\nabla S}(x_t,\delta) := \frac{1}{B}\sum_{b=1}^{B}\Phi_x * (x_t + K(r)\delta u_b)u_b, \tag{11}$$

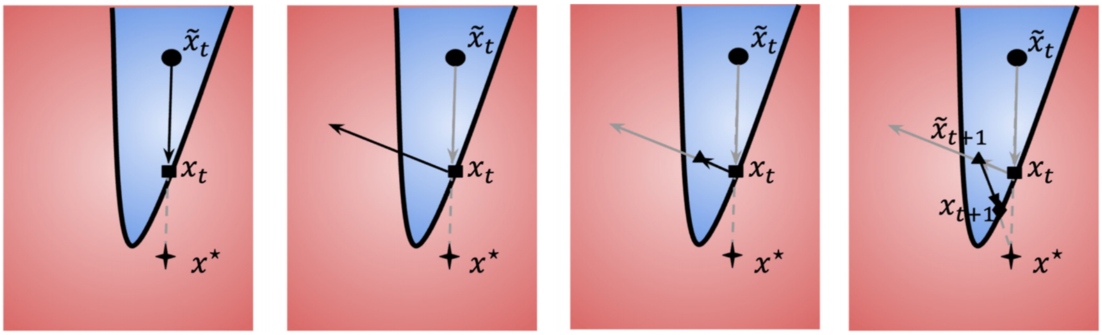


Fig. 4. The process of BAPP attack method as given by [33].

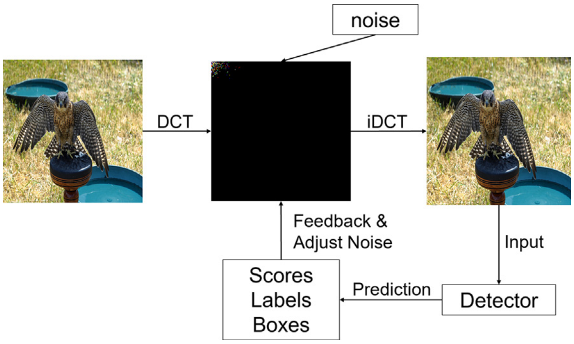


Fig. 5. The DCT-based boundary attack process.

where, $K(r) = [a_{ij}], a_{ij} = \begin{cases} 1, 1 \leqslant i,j \leqslant imgsize * \ r \\ 0, etc \end{cases}$. Here, r is the ratio which determines the proportion of low frequency. The bigger the r is, the higher frequency noise is added.

The feasibility and efficiency of the proposed mechanism are established through experimentation, which is discussed in the next section. The complete BA process in the frequency domain based on DCT is shown in Fig. 5. A binary approximation function is applied for projecting the image to the boundary as given by Algorithm 1. After getting the results, the DCT-based boundary attack method can be applied as given by Algorithm 2.

Algorithm 1 Binary-search for projecting the image to the boundary.

```
Require: upper bound $x_u$, lower bound $x_1$, threshold $\xi$
Require: Binary function $\varphi$, s.t. $\varphi(x_u)$=1, $\varphi(x_1)$=0
Ensure: Boundary point $x_m$ and $\varphi(x_m)$=1
function BINARY-SEARCH($x_u, x_1, \xi, \varphi$)
if dist($x_u, x_1$) $<= \xi$ then
    return $x_u$
else
    Set $x_m = (x_u + x_1) * \ 0.5$
    if $\varphi(x_m) = 1$ then
        Binary-search($x_m, x_1, \xi, \varphi$)
    else
        Binary-search($x_u, x_m, \xi, \varphi$)
    endif
endif
end function
```

Algorithm 2 The DCT-based boundary attack.

```
Require: The source image $x_0$, the targeted $y_0$, the decision function in object detector $\Phi$
Require: the binary threshold $\xi$, the high frequency ratio in DCT $r$, the estimated gradient step $\delta$
Require: The initial estimated gradient query times $B_0$, the greatest iteration times $T$
Ensure: the adversary example $x_{adv}$
The initialized example preprocessed by corner alignment $x_1$
$x_{adv}$=DCT($x_1$), $d$=dist($x_{adv}, x_0$)
for $t$ in 1,2,...,T-1do
    $x_{adv}$=Binary-search($x_{adv}, x_0, d * \ \xi, \Phi$)
    Compute $\nabla S(x_{adv}, \delta)$ via Eq. (2)
    update $\delta = \delta/\sqrt{t}, B = B * \sqrt{t}$
    Initial tangential stepsize $\epsilon = d/\sqrt{t}$
    while $\varphi(x_{adv} + \epsilon * \nabla S(x_{adv}, \delta)) = 0$ do
        $\epsilon = \epsilon/2$
    end while
    Set $x_{adv} = x_{adv} + \epsilon * \nabla S(x_{adv}, \delta)$, $d$=dist($x_{adv}, x_0$)
end for
Output: $x_{adv}$=Binary-search($x_{adv}, x_0, d * \ \xi, \Phi$)
```

6. Experiments and verification

The evaluation of the effectiveness of the proposed attack method has been done through extensive experimentation. The test environment is built using Ubuntu 16.04 running on an Intel Xeon CPU E5-2620 v4 @2.10 GHz system equipped with Nvidia GeForce RTX 2080Ti and 32 GB RAM. It also uses Python 3.5, ImportLib TensorFlow, NumPy, matplotlib, and Pillow for analysis. The source image for generating an adversarial example is randomly selected from COCO 2014 test set and we select the YOLO v3 as the offline victim black-box object detector.

The binary threshold ξ is set to 0.01, and the high-frequency ratio r is set to 1/8, based on the work in [34]. The estimated gradient step δ is set to 0.01, the initially estimated gradient query times B_0 are set to 100, and the greatest iteration times T

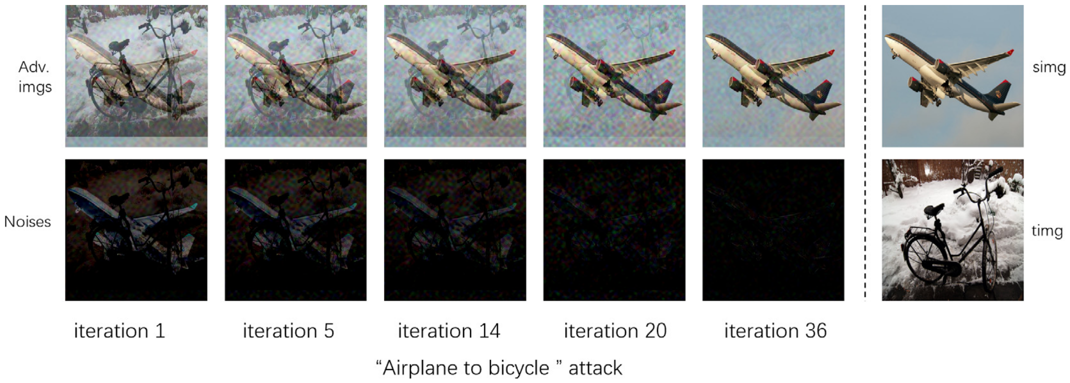


Fig. 6. The examples of DCT-based attack process and its noise.

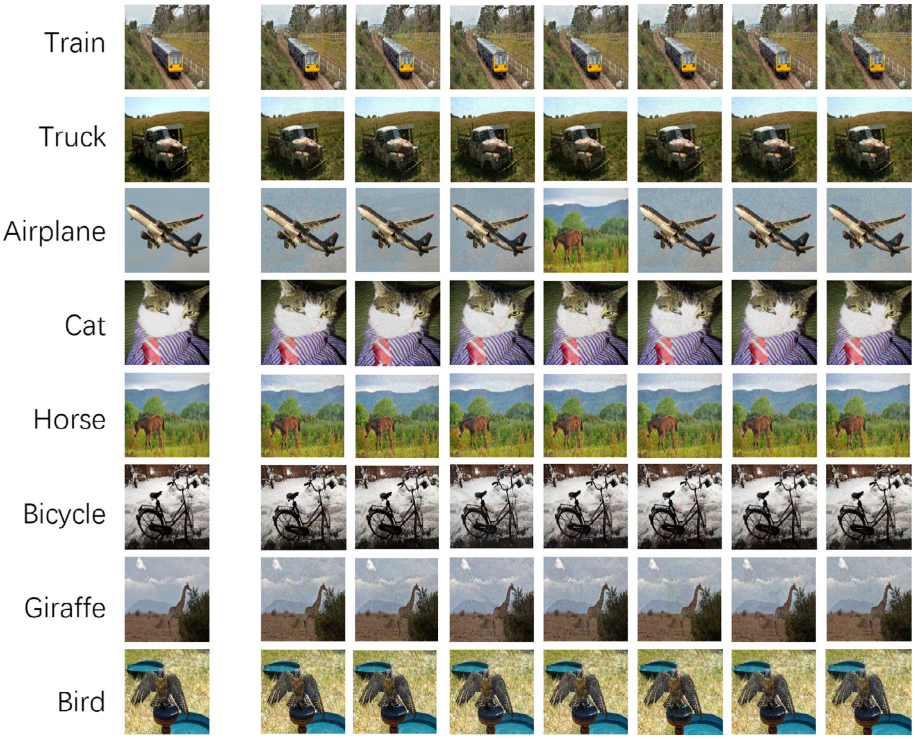


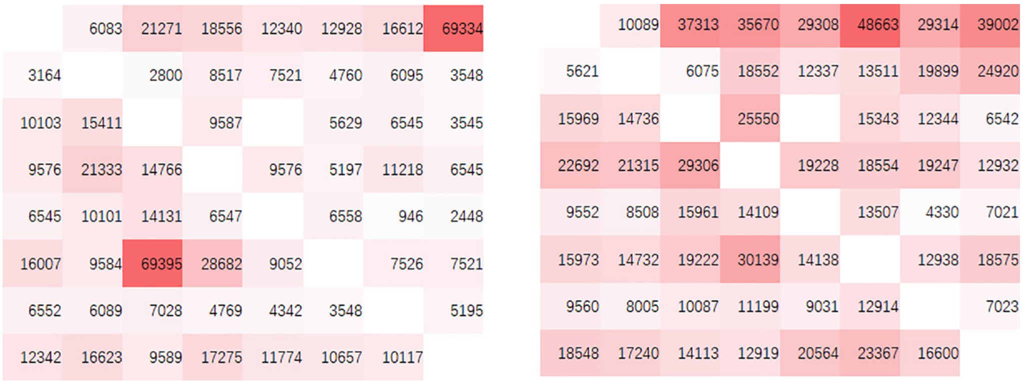
Fig. 7. The cross attack tests for eight images selected randomly.

are set to 100. These values are the same is in the BAPP proposal. The L_2 distance function is applied in our method similar to the majority of the attack methods in this domain.

6.1. Offline evaluation and verification

We first record the noise decreasing process of the adversary example as in Fig. 6. The airplane is the source image and the bicycle is the targeted image. The adversary examples are shown in the first line and the respective noise is shown in the second line. We can observe that the noise decreases sharply along with the iterations. The adversary example shows a good visual result at just the 36th iterations with very little noise. The noise applied in our method has slice effectiveness instead of pixel effectiveness, which is more smooth for the visual results.

The cross attack test is designed to verify the effectiveness and efficiency of the proposed DCT-based attack. We select eight images and implement cross attack as shown in Fig. 7. The stop condition is $L_2 < 20$ and the query times are recorded. The images selected only include one object respectively and only the targeted attack is implemented for simplification and highlighting the comparison results. In Fig. 7, the first image in each line is the source image, and the other seven images are



	6083	21271	18556	12340	12928	16612	69334
3164		2800	8517	7521	4760	6095	3548
10103	15411		9587		5629	6545	3545
9576	21333	14766		9576	5197	11218	6545
6545	10101	14131	6547		6558	946	2448
16007	9584	69395	28682	9052		7526	7521
6552	6089	7028	4769	4342	3548		5195
12342	16623	9589	17275	11774	10657	10117	

	10089	37313	35670	29308	48663	29314	39002
5621		6075	18552	12337	13511	19899	24920
15969	14736		25550		15343	12344	6542
22692	21315	29306		19228	18554	19247	12932
9552	8508	15961	14109		13507	4330	7021
15973	14732	19222	30139	14138		12938	18575
9560	8005	10087	11199	9031	12914		7023
18548	17240	14113	12919	20564	23367	16600	

Fig. 8. The thermogram of our DCT-based method(left) compared with BAPP(right).

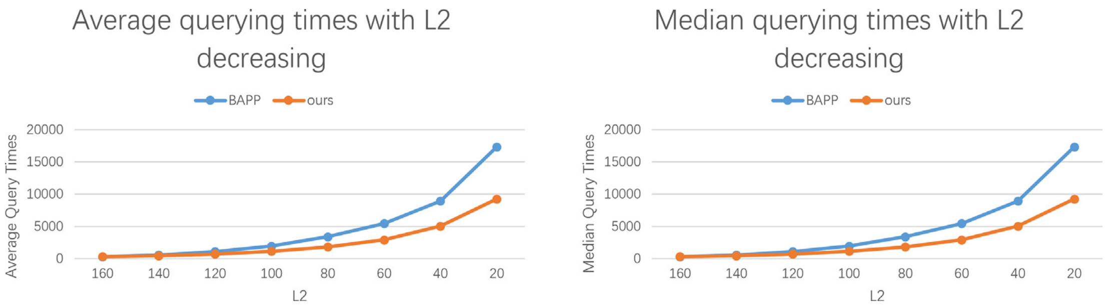
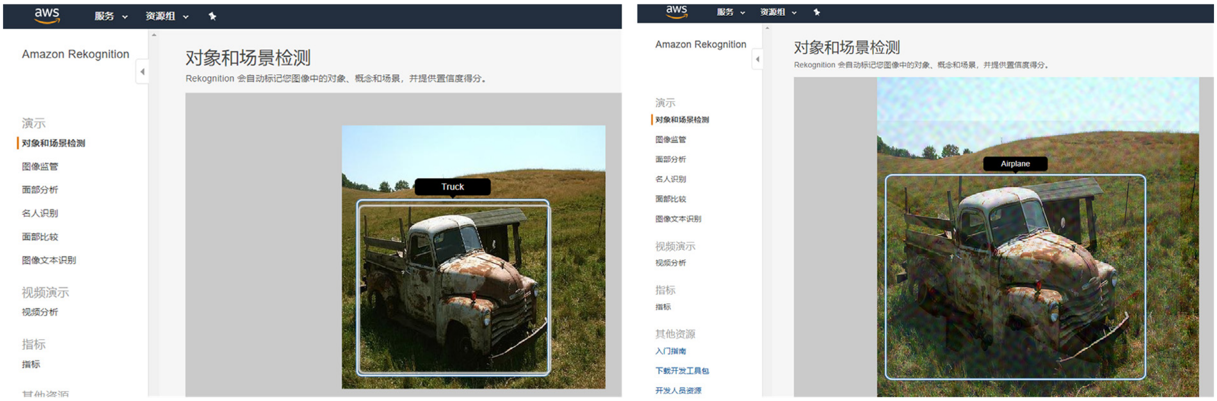


Fig. 9. The average query time and median query time of DCT-based method compared to BAPP.



对象和场景检测

Rekognition 会自动标记您图像中的对象、概念和场景，并提供置信度得分。

(a)

(b)

Fig. 10. The detection result of the source image *truck* and adversary example *airplane*.

the targeted adversary examples without the image itself. The attack is unsuccessful if the adversary example is shown as the targeted image (for example, the forth adversary image in the third line). There is only one failure in our attack test and the success rate is $55/56 = 98.2\%$.

We have also compared our DCT-based boundary attack method with BAPP. The query times are compared with the same L_2 threshold ($L_2 \leq 20$) as shown in Fig. 8. We can observe that most of the query times of our method are less than that of BAPP method in the thermogram.

Finally, we have extracted and computed the average and median query times from the thermogram and depicted the results in Fig. 9. It is can be observed that the average query time and median query times of the proposed novel DCT-based method are better than BAPP, for every time they achieve the same L_2 standard.

6.2. Attack results of online object detector

In this set of experiments, we have applied our attack method on an online object detector for verifying its effectiveness. The AWS Rekognition by Amazon has been used as the victim detector. AWS Rekognition is a subsystem in Amazon Cloud service and provides the services for image classification and object detection. The object and scene detection service is applied for our online attack test. We randomly select an image from the COCO test set. The detecting results of the source image and adversary example is shown in Fig. 10.

The source image is tagged correctly as a *truck* by AWS Rekognition as shown in Fig. 10(a). The targeted adversary example is an *airplane* generated by the proposed method from the truck source image. The adversary example generation stopped at the 29th iteration with the L_2 threshold, which is then input into the object and scene detection system. The detected result is shown in Fig. 10(b). We can observe that the object is framed and detected as an airplane as desired, and this verifies the effectiveness of our DCT-based boundary attack method.

7. Conclusion

Boundary attack methods are frequently used to evaluate the deep leaning systems for their effectiveness in identifying objects in given images. In this work, we present a novel black-box boundary attack method based on the discrete cosine transform. It can generate the adversarial example with a low L_2 distance and fewer query time. The test data shows that the proposed method can decrease the query times when it reaches the same L_2 threshold, or it can have lesser L_2 distance at the same level of query time as compared against other state-of-the-art attack methods in the black-box environment. The attack results on YOLO v3 offline and AWS Rekognition online verified its effectiveness and efficiency. A promising future work for this method is its experimental extension using different attack methods.

CRediT authorship contribution statement

Kuang Xiaohui: Methodology, Formal analysis. **Gao Xianfeng:** Software, Validation, Formal analysis, Resources. **Wang Lianfang:** Investigation. **Zhao Gang:** Data curation. **Ke Lishan:** Writing - review & editing. **Zhang Quanxin:** Conceptualization, Project administration, Supervision, Funding acquisition.

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.

Acknowledgment

This work is in part supported by National Natural Science Foundation of China (No. 61876019).

References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25, Curran Associates Inc, 2012, pp. 1097–1105.

[2] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR abs/1409.1556 (2015). URL: https://arxiv.org/abs/1409.1556..

[3] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, in: C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems 28, Curran Associates Inc, 2015, pp. 91–99.

[4] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27, Curran Associates Inc, 2014, pp. 3104–3112.

[5] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A.W. Senior, K. Kavukcuoglu, Wavenet: A generative model for raw audio, CoRR abs/1609.03499 (2016). URL: http://arxiv.org/abs/1609.03499..

[6] Z. Guan, Y. Zhang, L. Zhu, L. Wu, S. Yu, Effect: an efficient flexible privacy-preserving data aggregation scheme with authentication in smart grid, Sci. China Inf. Sci. 62 (3) (2019) 032103.

[7] Q. Zhang, H. Gong, X. Zhang, C. Liang, Y. Tan, A sensitive network jitter measurement for covert timing channels over interactive traffic, Multimedia Tools Appl. 78 (3) (2018) 3493–3509, https://doi.org/10.1007/s11042-018-6281-1.

[8] Z. Guan, X. Liu, L. Wu, J. Wu, R. Xu, J. Zhang, Y. Li, Cross-lingual multi-keyword rank search with semantic extension over encrypted data, Inf. Sci. 514 (2020) 523–540.

[9] T. Li, C. Gao, L. Jiang, W. Pedrycz, J. Shen, Publicly verifiable privacy-preserving aggregation and its application in IoT, J. Netw. Comput. Appl. 126 (2019) 39–44, https://doi.org/10.1016/j.jnca.2018.09.018.

[10] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, CoRR abs/1312.6199 (2013). URL: https://arxiv.org/abs/1312.6199..

[11] X. Gao, Y. Tan, H. Jiang, Q. Zhang, X. Kuang, Boosting targeted black-box attacks via ensemble substitute training and linear augmentation, Appl. Sci. 9 (11) (2019) 2286, https://doi.org/10.3390/app9112286.

[12] F. Guo, Q. Zhao, X. Li, X. Kuang, J. Zhang, Y. Han, Y. Tan, Detecting adversarial examples via prediction difference for deep neural networks, Inf. Sci. 501 (2019) 182–192, https://doi.org/10.1016/j.ins.2019.05.084.

[13] X. Wang, J. Li, X. Kuang, Y. Tan, J. Li, The security of machine learning in an adversarial setting: a survey, J. Parallel Distrib. Comput. 130 (2019) 12–23, https://doi.org/10.1016/j.jpdc.2019.03.003.

[14] A. Wu, Y. Han, Q. Zhang, X. Kuang, Untargeted adversarial attack via expanding the semantic gap, in: 2019 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2019, pp. 514–519, https://doi.org/10.1109/icme.2019.00095.
[15] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv:1412.6572 (2014)..
[16] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial examples in the physical world, CoRR abs/1607.02533 (2016). URL: https://arxiv.org/abs/1607.02533..
[17] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, J. Li, Boosting adversarial attacks with momentum, CoRR abs/1710.06081 (2017). URL: https://arxiv.org/abs/1710.06081..
[18] N. Papernot, P.D. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, A. Swami, The limitations of deep learning in adversarial settings, CoRR abs/1511.07528 (2015). URL: http://arxiv.org/abs/1511.07528..
[19] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 IEEE Symposium on Security and Privacy (SP), IEEE, 2017. https://doi.org/10.1109/sp.2017.49..
[20] N. Papernot, P.D. McDaniel, I.J. Goodfellow, S. Jha, Z.B. Celik, A. Swami, Practical black-box attacks against deep learning systems using adversarial examples, CoRR abs/1602.02697 (2016). URL: http://arxiv.org/abs/1602.02697..
[21] J. Su, D.V. Vargas, K. Sakurai, One pixel attack for fooling deep neural networks, IEEE Trans. Evol. Comput. 23 (5) (2019) 828–841, https://doi.org/10.1109/tevc.2019.2890858.
[22] W. Brendel, J. Rauber, M. Bethge, Decision-based adversarial attacks: reliable attacks against black-box machine learning models, CoRR abs/1712.04248 (2017). URL: https://arxiv.org/abs/1712.04248..
[23] A. Hassan, R. Hamza, H. Yan, P. Li, An efficient outsourced privacy preserving machine learning scheme with public verifiability, IEEE Access 7 (2019) 146322–146330, https://doi.org/10.1109/access.2019.2946202.
[24] T. Li, X. Li, X. Zhong, N. Jiang, C. zhi Gao, Communication-efficient outsourced privacy-preserving classification service using trusted processor, Inf. Sci. 505 (2019) 473–486, https://doi.org/10.1016/j.ins.2019.07.047.
[25] J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017. https://doi.org/10.1109/cvpr.2017.690..
[26] W. Wang, Y. Li, X. Wang, J. Liu, X. Zhang, Detecting android malicious apps and categorizing benign apps with ensemble of classifiers, Future Gen. Comput. Syst. 78 (2018) 987–994, https://doi.org/10.1016/j.future.2017.01.019.
[27] D.-D. Zhao, F. Li, K. Sharif, G.-M. Xia, Y. Wang, Space efficient quantization for deep convolutional neural networks, J. Comput. Sci. Technol. 34 (2) (2019) 305–317, https://doi.org/10.1007/s11390-019-1912-1.
[28] A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial machine learning at scale, CoRR abs/1611.01236 (2016). URL: http://arxiv.org/abs/1611.01236..
[29] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, C.-J. Hsieh, Zoo: zeroth order optimization based black-box attacks to deep neural networks without training substitute models, in: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, ACM Press, 2017, https://doi.org/10.1145/3128572.3140448.
[30] A. Ilyas, L. Engstrom, A. Athalye, J. Lin, Black-box adversarial attacks with limited queries and information, CoRR abs/1804.08598 (2018). URL: http://arxiv.org/abs/1804.08598..
[31] N. Ahmed, T. Natarajan, K. Rao, Discrete cosine transform, IEEE Trans. Comput. C 23 (1) (1974) 90–93, https://doi.org/10.1109/t-c.1974.223784.
[32] H. Wu, X. Meng, Y. Wang, Y. Yin, X. Yang, W. He, G. Dong, H. Chen, High compressive ghost imaging method based on discrete cosine transform using weight coefficient matching, J. Mod. Opt. 66 (17) (2019) 1736–1743, https://doi.org/10.1080/09500340.2019.1660816.
[33] J. Chen, M.I. Jordan, M.J. Wainwright, Hopskipjumpattack: A query-efficient decision-based attack, CoRR (2019). URL: https://arxiv.org/abs/1904.02144v4..
[34] C. Guo, J.S. Frank, K.Q. Weinberger, Low frequency adversarial perturbation, CoRR abs/1809.08758 (2018). URL: http://arxiv.org/abs/1809.08758..