

ORIGINAL RESEARCH

Open access

A Novel Dynamic Programming Approach for Generating Sparse Adversarial Examples in Deep Neural Networks

Fatima El Idrissi^{1*}, Samira Bennani¹, Youssef Benali², Ahmed Ben Ali³

Abstract

Recent investigations have demonstrated that deep neural networks are susceptible to subtly modified input data. This paper introduces a novel untargeted white-box adversarial attack termed the dynamic programming-based sub-pixel score method (SPSM) attack (DPSPSM). This technique modifies conventional gradient-based white-box adversarial methods, which are typically restricted by a predetermined Hamming distance, by incorporating a dynamic programming framework. The method is driven by a pixel score metric approach, the SPSM, which is developed herein. Unlike traditional gradient-based adversarial attacks that introduce nearly imperceptible modifications across the entire image, DPSPSM operates rapidly and maintains robustness by altering only a restricted number of input pixels. The introduced algorithm quantizes the gradient updates using a score computed for every pixel, which aggregates the contributions across all channels. Experimental outcomes indicate that the DPSPSM successfully misleads the model, achieving a success rate of 30.45% on the CIFAR-10 test set and 29.30% on the CIFAR-100 test set.

Keywords Adversarial attack strategies, Deep neural network vulnerabilities, Convolution neural network, Dynamic programming-based adversarial attack, Gradient-based adversarial attacks

*Correspondence:

Fatima El Idrissi
fatima.idrissi@uca.ma

¹ Department of Adversarial Machine Learning and Dynamic Programming, Faculty of Engineering, University of Marrakech, Marrakech, Morocco

² Department of Sparse Adversarial Generation, Faculty of Engineering, University of Fez, Fez, Morocco

³ Department of White-Box Attacks and Security, Faculty of Engineering, University of Casablanca, Casablanca, Morocco

Introduction

The escalating integration of machine learning (ML) has transformed contemporary technology, embedding its operations into daily routines. Deep learning (DL) models have expanded beyond conventional uses like algorithmic optimization and infrastructure planning, integrating themselves into daily human experiences [1] through virtual assistants [2], personalized web advertising [3, 4], recommendation algorithms [5-7], clinical imaging [8-10], and natural language processing [11, 12]. While their capacity for generalization and expressiveness underlies their success, these traits also lead to unexpected behaviors. The internal mechanisms of these networks are buried within intricate mathematical and statistical layers away from direct observation, rendering the identification of underlying biases and errors highly challenging.

Recent evidence highlighting the susceptibility [13-16] of DL models to meticulously engineered input data creates safety apprehensions regarding their deployment in high-stakes domains. Although these adversarial inputs trigger incorrect model predictions, they appear completely uncorrupted to human observers. Notably, numerous attack methodologies [13, 17-21] have been developed that successfully exploit these systems. Certain techniques operate effectively without

requiring prior knowledge of the target network's architecture. Furthermore, identical sets of minor input modifications can successfully degrade performance across multiple distinct models, revealing fundamental blind spots in our comprehension of how these systems operate [22-24]. It is hypothesized that ML models establish decision boundaries between different output classes to categorize incoming data, which introduces a degree of linearity within their expansive decision spaces [13].

Motivation

Adversarial attacks represent a critical dimension of security research, as they expose fundamental challenges within artificial intelligence and ML safety. Countering the widespread assumption that ML models operate primarily via non-linear mechanisms, Goodfellow et al. [13] established that the linear tendencies of these networks within high-dimensional spaces are precisely what create opportunities for adversarial exploitation. Although multiple defense strategies [18, 25-30] have been evaluated and have provided moderate protection, they remain vulnerable to more advanced and efficient attack methodologies.

Shamir et al. [31] introduced an adversarial generation technique based on a fixed Hamming distance constraint. While this approach is more computationally complex and slower than traditional gradient-based methods, such as those found in [13, 17], it presents an innovative strategy for modifying only a preselected arrangement of input components. This concept is further reinforced by the efficacy of the one-pixel attack [32] against state-of-the-art ML models, which raises concerns about the reliability of their achievements given their vulnerability to such minimal modifications.

In the field of digital image editing, Avidan and Shamir [33] demonstrated that successful image resizing necessitates considering geometric limitations alongside an evaluation and integration of the underlying image content. They introduced the seam carving technique, which facilitates content-aware image reduction and expansion.

The unusual vulnerabilities displayed by deep neural networks (DNNs) when subjected to adversarial attacks, such as those documented in [13, 17-21], have motivated researchers to leverage rapid gradient-based approaches while simultaneously restricting the scope of the applied perturbations. This premise is validated by findings in [13, 31, 32] and is further corroborated by the evaluations conducted in this study. Our objective is to design a straightforward yet potent adversarial attack that requires modifying only a few pixels, akin to [31], while matching the rapid execution speed found in [13].

In this study, we introduce a white-box adversarial attack that leverages dynamic programming (DP). We first outline the principles of pixel value manipulation, leading to the formulation of the sub-pixel score method (SPSM), which functions as a pixel score metric. We detail the attack framework linked to this score, subsequently presenting the dynamic programming-based sub-pixel score method attack (DPSPSM), which integrates the SPSM with DP. While this research targets the generation of perturbations for image inputs within convolutional neural networks, subsequent investigations could extend these concepts to alternate architectures, such as text-oriented recurrent neural networks. The DPSPSM generates perturbations rapidly compared to the method in [32], yet alters only a limited number of pixels compared to the approach in [13]. The introduced attack does not outperform these state-of-the-art methods in absolute performance metrics; rather, it synthesizes their respective merits.

Contributions

This paper introduces a novel untargeted white-box adversarial attack, known as the dynamic programming-based sub-pixel score method (DPSPSM) attack. This method adapts traditional gradient-based white-box adversarial frameworks by constraining the search space within a fixed Hamming distance through a dynamic programming architecture. The DPSPSM attack utilizes dynamic programming to locate and modify a minimal collection of pixels, specifically targeting a single vertical seam within the image. This strategy markedly decreases computational costs compared to techniques that rely on intensive iterative procedures or full-image modifications. For example, the one-pixel attack [32] utilizes differential evolution (DE) to determine the optimal pixel for perturbation, a process that is computationally demanding and requires

many iterations to converge. Conversely, DPSPSM couples a gradient-driven mechanism with dynamic programming, enabling a more direct and computationally efficient execution. The authors present the SPSM to construct a score metric map for every pixel in the image, which subsequently guides the DPSPSM attack. This attack serves as a gradient-based, untargeted white-box method restricted to the space of visual perturbations.

The innovative nature of the DPSPSM attack stems from its focused targeting of a single vertical seam, which causes minimal visual disruption while significantly degrading model accuracy. This subtle modification is essential for analyzing the vulnerabilities of DNNs to adversarial samples. The reality that humans can seamlessly identify the correct content of perturbed images while deep neural networks (DNNs) fail to do so underscores a profound divergence between human and machine visual processing systems. This variance challenges the dependability of contemporary DNN architectures and points toward future research pathways centered on constructing models that better replicate human visual processing. Such developments could markedly bolster the defense of DNNs against adversarial threats.

The primary contribution of this research is the DPSPSM, which produces perturbations rapidly, unlike the computationally demanding and highly iterative method in [32]. Furthermore, while the FGSM [13] exhibits a superior success rate, it accomplishes this by altering the entire image, representing a more discernible and less refined strategy than the localized approach of DPSPSM. This distinction is vital for formulating defensive mechanisms against adversarial threats, as it indicates that prioritizing localized, less conspicuous modifications could provide a more potent means of defense. The proposed method does not surpass the success rates of state-of-the-art techniques like FGSM [13] or the One Pixel Attack [32], but it fuses their unique advantages into a novel, fast, and constrained methodology. This work is highly relevant to the adversarial machine learning community, as the arguments outlined and the corresponding minimal changes enforced by DPSPSM show high efficacy in altering model predictions, as detailed in **Table 1**. Additionally, the computational economy of DPSPSM relative to the One Pixel Attack highlights its viability for scalability and deployment in resource-limited settings. This aspect of the research expands the ongoing dialogue regarding adversarial machine learning and supports the design of more efficient, covert attack strategies, thereby advancing our comprehension of the vulnerabilities inherent in deep neural networks (DNNs).

Related Work

Adversarial attack

Machine learning models and deep neural networks (DNNs) have advanced significantly, augmenting critical practical applications involving images [34, 35], videos, speech [36-38], voice [39, 40], text [41, 42], neural network dynamics [43], and various other domains [44]. Despite these advancements, these networks remain highly susceptible and vulnerable to diverse adversarial attacks [13, 14, 16, 17, 32, 45-51]. Furthermore, the engineered adversarial examples often possess robust transferability across different network architectures [23, 24]. Although multiple defensive strategies have been introduced, such as those in [18, 25-30], the robustness of newer attack mechanisms like [18, 52] has largely rendered them ineffective.

Gradient based adversarial approach

Goodfellow *et al.* [13] introduced the Fast Gradient Sign Method (FGSM), a gradient-based attack rooted in the concept of linearity within high-dimensional decision spaces. This methodology requires a forward pass through the input data to compute the gradient at each constituent step, subsequently integrating a fraction of this gradient to produce adversarial perturbations. Gradient-based attacks [13, 17, 18, 22, 25, 27, 53] feature rapid computational speeds because they only necessitate a single forward pass over the model; however, they modify every pixel within the target image. The proposed DPSPSM leverages this rapid gradient-based framework while concurrently restricting the number of altered pixels.

Fixed length disturbance based adversarial approach

Su *et al.* generated adversarial examples using Differential Evolution (DE) by modifying only a single pixel [31], bypassing the challenges of perceptibility measurement and expanding upon the principle of fixed-length disturbances [31, 54, 55]. The potency of this attack is demonstrated by the difficulty in identifying the introduced perturbation. Nonetheless, DE is computationally intensive and demands numerous iterations to achieve convergence. Conversely, DPSPSM integrates the concept of localized, fixed-length modifications while preserving the rapid computational execution characteristic of gradient-based frameworks.

Content aware image resizing

Avidan and Shamir demonstrated that efficient image resizing should extend beyond merely adhering to geometric constraints. To address this, they introduced Seam Carving [33], a content-aware image resizing methodology applicable to both expansion and reduction. This technique highlights the utility of dynamic programming within digital image manipulation. In our work, we adapt this dynamic programming approach to establish the underlying structural modifications required in the image for the DPSPSM framework.

Relevance

DPSPSM aims to fuse the robustness and celerity of the gradient-based framework [13] with the efficacy of a predefined Hamming distance constraint [31], utilizing the dynamic programming principles established in [33]. Although FGSM [13] is effective and computationally elementary, it still adjusts every pixel of the input image. In contrast, DPSPSM seeks to manipulate only a predetermined set of pixels within the input image to successfully deceive the target models. Furthermore, DPSPSM differs fundamentally from the one-pixel attack [32]; the latter relies on differential evolution, whereas DPSPSM utilizes gradient-based calculations coupled with the dynamic programming framework outlined in [33]. While the black-box nature of the one-pixel attack provides resilience, the computational overhead required by DE is substantial. The gradient-based process of DPSPSM, although a white-box attack, necessitates only a direct forward pass.

The objective of DPSPSM is not to challenge the aforementioned state-of-the-art attacks, but rather to substantiate the efficacy and simplicity of enforcing modifications on a strictly constrained number of pixels via a gradient-based scheme that synthesizes the advantages of both worlds.

Materials and Methods

Introduction

The generation of adversarial images can be framed as a constrained optimization problem. The goal is to introduce perturbations to the pixels of an original image such that the visual presentation remains fundamentally unchanged, yet the model's predicted class label shifts to an incorrect category.

Two primary approaches are typically employed to achieve this objective:

1. **Fixed L0 norm (or Hamming distance):** This strategy restricts the total number of pixels available for perturbation, while placing no constraints on the magnitude of those individual perturbations.
2. **Fixed LK norm:** This strategy permits adjustments across all pixels, but enforces a global constraint on the cumulative magnitude of the applied disturbances.

This study adopts the **Fixed L0 norm** methodology, meaning the generation process is bound by a constraint on the total number of pixels permitted to undergo modification.

Let $J(x, \theta, y)$ denote the cost function utilized during neural network training to achieve accurate image classification, where x represents the input pixel vector (corresponding to the original source image), θ represents the learned network weights, and y denotes the ground-truth output label.

Following network training, the optimized weights θ remain static, while we adjust the input image pixels to introduce perturbations that shift the model prediction as far as possible from its authentic value, thereby generating adversarial inputs. Stated another way, the probability of the modified image being assigned to its true class must be minimized. Because this work introduces an untargeted attack, the specific incorrect label assigned to the perturbed image is irrelevant, provided it differs from the true class label.

This formulation is equivalent to maximizing the original loss function $J(x, \theta, y)$ treating x as the parameter of interest instead of θ (which remains constant), subject to a constraint on the number of pixels allowed to change in x , as formalized in [Eq. 1](#):

$$\left(\begin{array}{l} \text{maximise } J(x, \theta) \\ \text{subject to } H(x, x_0) \leq K \end{array} \right) \quad (1)$$

where:

- x_0 is the original input image undergoing perturbation.
- $H(x, x_0)$ is the Hamming distance between x and x_0 , representing the number of pixels in which they differ.
- K is a predefined upper bound restricting the maximum number of pixels permitted for perturbation.

Perturbation rule: 0–255 approach

To establish a transparent foundation, we initiate the discussion of our perturbation methodology using a first-order Taylor expansion of the loss function. As posited by Goodfellow *et al.* [13], the decision boundaries of models within high-dimensional spaces are piecewise linear, and this characteristic serves as the foundation for the existence of adversarial examples. Our objective cost function $J(x, \theta, y)$ can be approximated via the first-order term of the Taylor series expansion, as expressed in [Eq. 2](#):

$$J(x, \theta, y) \approx J(x_0, \theta, y) + (x - x_0)^T \nabla_x J(x_0, \theta, y) \quad (2)$$

Given that (x_0, θ, y) (the initial prediction cost) is a fixed constant, maximizing (x, θ, y) , simplifies to maximizing the term $(x - x_0)^T \nabla_x J(x_0, \theta, y)$, subject to the primary constraint of K pixels. This operation essentially calculates the weighted sum of sub-pixel perturbations scaled by their respective sub-pixel gradients.

Our perturbation framework depends directly on the gradient, ∇ , of the cost function (x, θ, y) computed with respect to the sub-pixels. By operating under the assumption that the perturbation rule is derived from a first-order Taylor expansion of the loss function, the monotonicity of the gradient relative to the sub-pixel is inherently satisfied.

We employ a straightforward gradient-driven perturbation technique. A sub-pixel is defined as an individual channel component of a given pixel. Consequently, in an RGB image, each pixel comprises a Red (R), Green (G), and Blue (B) sub-pixel, which collectively define the pixel's color presentation. By adjusting a single sub-pixel or all sub-pixels within a pixel, we directly alter the resulting illumination of that pixel.

The gradient, ∇ , yields the trajectory required to navigate the N -dimensional space (N being the total number of sub-pixels in the input image) to achieve the steepest increase (or decrease) in the loss. The sign of the gradient relative to

each input sub-pixel dictates whether that specific sub-pixel value must be increased or decreased to elevate the prediction cost.

Because we operate under a fixed L0 norm constraint and do not restrict the perturbation magnitude of an individual sub-pixel, our methodology drives the value completely to its operational extremes: either up to 255 or down to 0. This directional assignment depends entirely on the sign of the gradient, reflecting whether the gradient indicates an increase or decrease for that specific sub-pixel value.

We utilize the notation i_c to index the sub-pixel corresponding to the c^{th} channel of the i^{th} pixel.

Accordingly, we formalize the updated value of the sub-pixel, $x(i_c)$, as defined in **Eq. 3**:

$$y(i_c) = \begin{cases} 255, & \text{if } \text{sgn}(\nabla(i_c)) = 1 \\ 0, & \text{if } \text{sgn}(\nabla(i_c)) = -1 \\ x(i_c), & \text{if } \text{sgn}(\nabla(i_c)) = 0 \end{cases} \quad (3)$$

where:

- (i_c) is the ultimate value of the i_c^{th} sub-pixel.
- (i_c) is the initial value of the i_c^{th} sub-pixel.
- $\nabla(i_c)$ is the calculated gradient of the i_c^{th} sub-pixel.
- (x) is the signum function.

The perturbation rule detailed in **Eq. 3** serves as the core mechanism within the DPSPSM framework for modifying pixel values.

Sub-pixel scoring method

A Preliminary score measure for pixel selection

Because our attack is restricted to modifying a maximum of K pixels, as formulated in (1), we require a strategy to assign a score to each sub-pixel. This scoring mechanism will subsequently enable the optimal selection of pixels for the DPSPSM framework.

Derived from the perturbation rule established in (3), a straightforward yet reasonable strategy for scoring each sub-pixel, (i_c) , is defined as follows:

$$D(i_c) = \begin{cases} 255 - x(i_c), & \text{if } \text{sgn}(\nabla(i_c)) = 1 \\ x(i_c), & \text{if } \text{sgn}(\nabla(i_c)) = -1 \\ 0, & \text{if } \text{sgn}(\nabla(i_c)) = 0 \end{cases} \quad (4)$$

where:

- (i_c) represents the score computed for the sub-pixel.
- $\nabla(i_c)$ is the gradient corresponding to the sub-pixel.
- (x) is the signum function.

The value (i_c) serves as a metric indicating the maximum allowable variation permitted for the sub-pixel, conditioned on the sign of the gradient ∇ . Intuitively, sub-pixels capable of undergoing greater degrees of modification according to the gradient—thereby driving a larger increase in the overall prediction loss—are assigned higher priority.

Improving the score

Adding gradient magnitudes

The initial scoring technique formulated in (4) considers only the direction of the gradient and the allowable range of sub-pixel perturbation. However, it becomes apparent that the actual magnitude of the gradient can serve as an effective heuristic, yielding richer informational value than what is conveyed by the sign alone.

Consequently, we refine the score for the (i_c) sub-pixel, i_c^{th} , in **Eq. 5**:

$$D(i_c) = \begin{cases} |\nabla(i_c)| * (255 - x(i_c)), & \text{if } \text{sgn}(\nabla(i_c)) = 1 \\ |\nabla(i_c)| * x(i_c), & \text{if } \text{sgn}(\nabla(i_c)) = -1 \\ 0, & \text{if } \text{sgn}(\nabla(i_c)) = 0 \end{cases} \quad (5)$$

Vectorizing the Score

The maximum allowable perturbation magnitude, which functioned as our preliminary score metric in (4), can be unified into a single mathematical expression, $F(i_c)$, for instances where the sign of the gradient is either positive or negative (excluding zero), as shown in (6):

$$F(i_c) = \frac{(1 + \text{sgn}(\nabla(i_c)))(255 - x(i_c)) + (1 - \text{sgn}(\nabla(i_c)))x(i_c)}{2} \quad (6)$$

To refine this further, two objectives must be met: pixels possessing a gradient of zero must be prevented from yielding a non-zero value through this expression, and the magnitude of the gradient must be integrated. Both constraints are satisfied by multiplying the expression by the single term $|\nabla(i_c)|$, which represents the absolute value of the gradient.

This yields the unified expression formalized in **Eq. 7**:

$$D(i_c) = |\nabla(i_c)| * F(i_c) \quad (7)$$

Which can be expanded into **Eq. 8**:

$$D(i_c) = |\nabla(i_c)| \left(\frac{(1 + \text{sgn}(\nabla(i_c)))(255 - x(i_c)) + (1 - \text{sgn}(\nabla(i_c)))x(i_c)}{2} \right) \quad (8)$$

This expression simply computes the product of the gradient magnitude and the maximum allowable perturbation value for the i_c^{th} sub-pixel. Crucially, this formulation can be vectorized to accelerate the score computation for every sub-pixel across the input image matrix.

Combining Sub-Pixel Scores

Given that all C sub-pixels of a pixel (C denoting the total number of color channels) are interdependent and mapped to the same spatial coordinate, they cannot be selected independently. Ultimately, the modifications introduced by our attack must manifest at the pixel level rather than across isolated sub-pixels. For the i^{th} pixel, choosing either a single sub-pixel or all C sub-pixels for the DPSPSM implies that the pixel itself has been selected. Therefore, the selection algorithm must operate on entire pixels rather than sub-pixels.

To facilitate this pixel-level operation, we assign a structural score to each pixel by computing the mean of all its constituent C sub-pixel scores. This averaging approach ensures that every individual sub-pixel contributes equally to the final selection metric of its parent pixel.

Consequently, for the i^{th} pixel, we formalize the score $\text{Score}(i)$ in **Eq. 9**:

$$D(i) = \frac{\sum_{c=0}^C |\nabla(i_c)|((1 + \text{sgn}(\nabla(i_c)))(255 - x(i_c)) + (1 - \text{sgn}(\nabla(i_c)))x(i_c))}{2C} \quad (9)$$

where:

- (i_c) denotes the sub-pixel corresponding to i_c^{th} sub-pixel.
- $\nabla(i_c)$ signifies the gradient computed for the i_c^{th} sub-pixel.
- (x) is the signum function.
- C represents the total number of channels.

The metric produced via **Eq. 9** is defined as the SPSM score of the i^{th} pixel in the image. This score is subsequently utilized to guide pixel selection within the proposed DPSPSM attack framework.

We also assessed a basic adversarial approach based exclusively on the SPSM metric—termed the SPSM attack—by sorting pixels in descending order according to their SPSM scores, selecting the top K instances, and modifying these K pixels via the transformation function detailed in (3). The model evasion success rates corresponding to varying values of K for the SPSM attack on the CIFAR-10 dataset [35] are provided in **Table 1**. Setting $K=30$ on the CIFAR-100 dataset [35] produced a success rate of 35%. As these findings demonstrate, localized clustering of perturbations can occur under a greedy approach, which risks degrading the image to a degree that compromises human recognition of the underlying subject.

Dynamic programming based SPSM attack—DPSPSM

As established in the preceding section, a basic attack strategy that greedily selects the top K pixels according to their raw SPSM scores can scatter perturbations haphazardly across the image or concentrate them within a narrow zone. This compromises semantic clarity and harms human perception. A more structured and localized arrangement is required to constrain the perturbations, preserving the visual characteristics and semantic integrity of the image.

We employ dynamic programming to build the baseline path for these perturbations, utilizing a strategy adapted from the seam carving methodology [33] designed for content-aware image resizing. A vertical seam is defined as a contiguous path of pixels extending from the top row of an image to the bottom row, where the path may shift left or right by a maximum of one pixel or maintain its column position when moving between adjacent rows. Consequently, for any given

pixel, the algorithm evaluates at most the three adjacent pixels located directly above it. While our experiments focus on generating vertical seams, horizontal seams can be constructed using an analogous approach. This seam minimizes the visual footprint across the image space while systematically exploiting the vulnerabilities of deep learning models within a structurally coherent format. The resulting seam isolates exactly one pixel per row.

The operational pipeline for seam generation is organized as follows. First, the Sub-Pixel Score Matrix (SPSM) of the target image is calculated using [Eq. 9](#). Next, dynamic programming is applied directly to this generated SPSM to isolate the optimal path.

The coordinates of the i^{th} pixel are represented as (i_x, i_y) , where i_x denotes the row index and i_y denotes the column index. The notations i and (i_x, i_y) are used interchangeably to refer to the same pixel location.

Therefore, we establish the matrix constraints based on the spatial dimensions of the image:

$$\begin{aligned} 1 &\leq i_x \leq H \\ 1 &\leq i_y \leq W \end{aligned}$$

where:

- H represents the total height of the input image.
- W represents the total width of the input image.

We now define the dynamic programming matrix, DP , and the parent tracking matrix, P , utilized within the DPSPSM attack framework. Let (i) denote the dynamic programming accumulative score computed for the i^{th} pixel. The value of (i) is determined using the recurrence relation defined in [Eq. 10](#):

$$DP(i) = DP(i_x, i_y) = \begin{cases} D(i), & \text{if } i_x = 1 \\ D(i) + \max(DP(i_{x-1}, i_{y-1}), DP(i_{x-1}, i_y), DP(i_{x-1}, i_{y+1})), & \text{otherwise} \end{cases} \quad (10)$$

where:

- (i) is the localized SPSM score of the i^{th} pixel derived from [Eq. 9](#).

Thus, (i) aggregates the baseline SPSM score of the i^{th} pixel and the maximum cumulative DP value from its three adjacent neighbors in the preceding row. These three adjacent neighbors share either an edge or a vertex with the target pixel. Consequently, for a pixel situated at row i_x and column i_y , the valid adjacent neighbors are located at row $i_x - 1$ across columns $i_y - 1$, i_y , and $i_y + 1$. Because pixels in the first row ($i_x = 1$) lack an overarching row, their initial scores are set equal to their baseline SPSM scores.

We further define $P(i)$ to track the specific adjacent pixel in the row directly above the i^{th} pixel that contains the maximum associated value. Accordingly, $P(i)$ is calculated using [Eq. 11](#):

$$P(i) = \underset{j}{\operatorname{argmax}} (DP(j) \mid j_y \in \{i_{y-1}, i_y, i_{y+1}\}) \quad (11)$$

Therefore, [Eqs. 10 and 11](#) are evaluated sequentially to construct the full profiles for (i) and $P(i)$.

After completing the forward traversal across all rows, the optimal seam designated for adversarial perturbation is determined by evaluating the final row of the image. The pixel yielding the maximum cumulative value for (i) is selected where $i_x = height$, and each of its constituent sub-pixels is modified according to the transformation rule formalized in (3).

Next, the algorithm backtracks to the row immediately above this position. We isolate the parent pixel denoted by (i) for the current pixel, and perturb its sub-pixels using the same strategy outlined in (3). The tracking reference i is then replaced by its parent, (i) . This backtracking process is repeated iteratively until the algorithm reaches the initial row ($i_x = 1$).

This operational workflow can be structured as follows:

1. Select the initial pixel i such that (i) is maximized subject to the constraint $i_x = height$.
2. While $i_x > 0$:
 - Perturb the sub-pixels of the current pixel i via [Eq. 3](#).

Update the pixel index to its parent: $i = (i)$

Through this sequence, we construct a cohesive vertical seam that extends from the upper boundary of the image down to its baseline. This identical procedure can be configured symmetrically to yield a horizontal perturbation seam.

Our methodology minimizes the total number of modified pixels, satisfying the constraints discussed by Shamir *et al.* [31], while executing via the rapid gradient-based framework introduced by Goodfellow *et al.* [13] and adapting the dynamic programming seam carving principles developed by Avidan and Shamir [33].

The systemic pipeline for the DPSPSM framework is illustrated in [Figure 1](#) and formalized in [Algorithm 1](#).

Input: CIFAR 10 image dataset containing image X and labels Y , CNN VGG16 network

Output: Perturbed Images

```

Get_Subpixel_Score_Matrix( $X, G$ ) :
     $P \leftarrow \text{Sign}(G)$ 
     $\text{Norm}_X \leftarrow X/255$ 
     $G \leftarrow \text{normalize}(G)$ 
     $\text{Subpixel\_Score} \leftarrow (\text{abs}(G) * ((1 + P) * (1 - G) + (1 - P) * G))/2$ 
     $\text{Subpixel\_Score\_Matrix}$ 
         $\leftarrow \text{Subpixel\_Score}[0, :, :, 0] + \text{Subpixel\_Score}[0, :, :, 1]$ 
         $+ \text{Subpixel\_Score}[0, :, :, 2]$ 
    return  $\text{Subpixel\_Score\_Matrix}$ 

Calculate_Max_Subpixel_Score( $\text{SSM}$ ) :
     $X_{\text{matrix}} \leftarrow \text{SSM}$ 
     $\text{Parent} \leftarrow [0 : 0]$ 
    for  $\text{row}$  in  $1 \dots X_{\text{matrix\_shape}}[0]$  :
    for  $\text{col}$  in  $1 \dots X_{\text{matrix\_shape}}[1]$  :
         $\text{Index} \leftarrow \text{argmax}(X_{\text{matrix}}[\text{row} - 1, \text{col} - 1 : \text{col} + 2])$ 
         $\text{Parent}[\text{row}, \text{col}] \leftarrow \text{Index} + \text{col} - 1$ 
         $X_{\text{matrix}}[\text{row}, \text{col}] += X_{\text{matrix}}[\text{row} - 1, \text{Index} + \text{col} - 1]$ 
    return  $X_{\text{matrix}}, \text{Parent}$ 

 $X_{\text{perturbed}} \leftarrow []$ 
1. for  $i$  in  $1 \dots N$  :
2.    $G \leftarrow \text{gradient}_{\text{VGG16}}(X[i], Y[i])$ 
3.    $\text{SSM} \leftarrow \text{Get\_Subpixel\_Score\_Matrix}(X[i], G[i])$ 
4.    $X[i]_{\text{matrix}}, \text{Parent}[i] \leftarrow \text{Calculate\_Max\_Subpixel\_Score}(\text{SSM})$ 
5.    $P \leftarrow \text{sign}(G[i])$ 
6.    $X[i]_{\text{perturbed}} \leftarrow X[i]$ 
7.    $j \leftarrow \text{argmax}(X[i]_{\text{matrix}}[-1])$ 
8.
9.   for  $\text{row}$  in  $X[i]_{\text{shape}}[0] \dots 1$  :
10.     $X[i]_{\text{perturbed}}[\text{row}, j, 0] \leftarrow 255$  if  $P[\text{row}, j, 0] == 1$  else  $0$ 
11.     $X[i]_{\text{perturbed}}[\text{row}, j, 1] \leftarrow 255$  if  $P[\text{row}, j, 1] == 1$  else  $0$ 
12.     $X[i]_{\text{perturbed}}[\text{row}, j, 2] \leftarrow 255$  if  $P[\text{row}, j, 2] == 1$  else  $0$ 
13.     $j = \text{Parent}[i][\text{row}, j]$ 
14. return  $X_{\text{perturbed}}$ 

```

Algorithm 1. Dynamic programming Based SPSM attack-DPSPSM

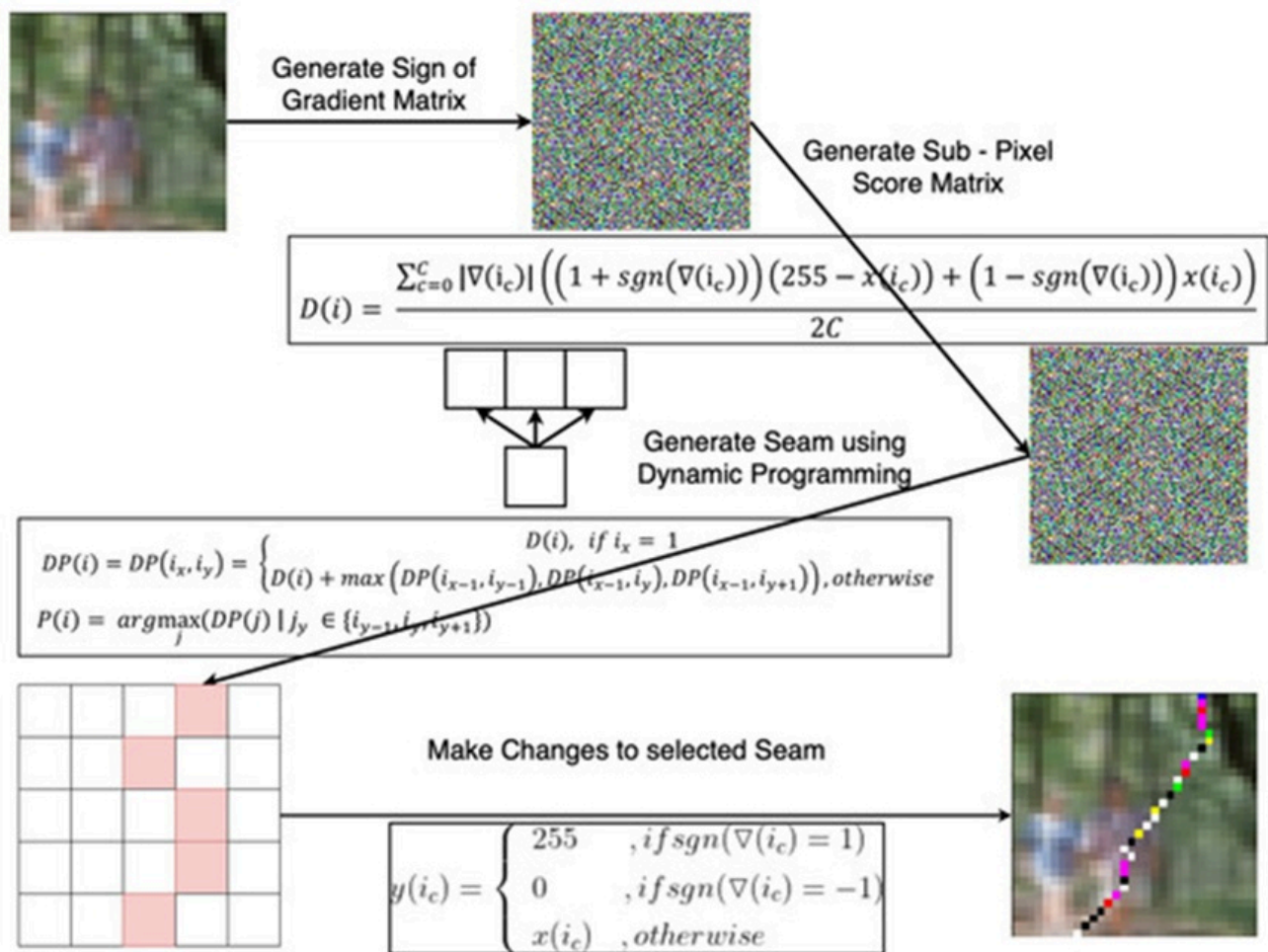


Figure 1. Flow describing the Dynamic programming based SPSM attack—DPSPSM.

The sequential breakdown of the operations in **Figure 1** is detailed below:

- **Step 1: Generate Sign of Gradient Matrix:** This initial phase involves computing the gradient of the loss function with respect to every sub-pixel in the image. This gradient vector dictates the precise directional adjustment required for each sub-pixel to drive the steepest increase in the loss function.
- **Step 2: Generate Sub-Pixel Score Matrix:** In this phase, a score matrix is constructed for every sub-pixel across the image. However, because all constituent sub-pixels of a single pixel (such as the red, green, and blue color channels in a standard RGB image) are physically linked to the same spatial coordinate, they cannot be evaluated independently. This dependency is critical since the ultimate adjustments introduced by the attack manifest as alterations to full pixels rather than isolated sub-pixels. The metric derived via **Eq. (9)** establishes the SPSM score for the i -th pixel, which subsequently guides the pixel selection mechanism within the proposed DPSPSM attack.
- **Step 3: Generate Seam Using Dynamic Programming:** During this stage, dynamic programming is executed to discover a continuous path of pixels—a seam—stretching from the top edge of the image to the bottom. The optimization objective is to isolate a path that maximizes the cumulative sum of the sub-pixel scores along the traversed route.

Seam generation process

We leverage dynamic programming to construct the structural trajectory of the perturbations. A vertical seam is defined as a sequence of pixels extending from the top row of the image to the bottom row, shifting left or right by at most one

column or remaining within the same column during the transition between adjacent rows. For any given pixel, the algorithm evaluates a maximum of three preceding neighboring pixels located directly above it.

1. **Initialization:** For the first row ($i_x = 1$), $DP(i)$ is set directly to the baseline SPSM score of the pixel, as no overriding rows exist.
2. **Recurrence Relation:** For every subsequent row, $DP(i)$ is computed via [Eq. \(10\)](#) as the sum of the current pixel's SPSM score from [Eq. \(9\)](#) and the maximum cumulative DP value among the three adjacent pixels in the row immediately above.
3. **Parent Tracking:** The matrix $P(i)$ utilizes [Eq. \(11\)](#) to store the spatial index of the specific pixel in the preceding row that produced the maximum cumulative DP value.

Seam selection

Once all image rows have been processed, the final perturbation path is isolated by evaluating the pixels in the bottom row. The pixel exhibiting the largest $DP(i)$ value in the last row is chosen, and its sub-pixels are modified according to [Eq. \(3\)](#). We then retrace the optimal path back up through the image by referencing the parent matrix $P(i)$:

1. **Start from the Bottom:** Identify the pixel in the final row that yields the maximum value for $DP(i)$.
2. **Trace Back:** Follow the pointers stored in $P(i)$ to navigate backward to the top row, applying the sub-pixel perturbation rule to each selected pixel along the path.

This execution scheme ensures that the resulting seam spans seamlessly from the top boundary to the bottom boundary of the image, selecting exactly one pixel per row. The framework can be adapted to produce a horizontal seam if required. This strategy minimizes visible alterations while effectively exploiting the underlying vulnerabilities of deep learning models.

- **Step 4: Make Changes to Seam Selection:** Once the optimal seam path has been isolated, the sub-pixels belonging to the pixels along this seam are perturbed based on the calculated scores using the transformation rule in [Eq. \(3\)](#).

By executing this sequence of steps, the algorithm successfully constructs a structured perturbation in the shape of a continuous seam, maximizing adversarial degradation while preserving visual fidelity. Randomness within the perturbations is maintained because the specific pixels selected for modification are not rigidly predetermined; instead, they are dynamically isolated for each image based on its unique mathematical characteristics. This ensures that the applied noise is custom-tailored to the image. Furthermore, the magnitude of pixel adjustment is inherently dependent on the algorithmic output, enabling adaptive perturbations that maximize adversarial efficacy.

The source code for the DPSPSM implementation can be accessed at: <https://colab.research.google.com/drive/1W1KV3gJrGewuz5SzYyD8qg3LxuqVhsyY?usp=sharing#scrollTo=0ty48n-HhNcg> (accessed on 15 January 2024).

Table 1 details the empirical success rates of the DPSPSM attack evaluated on the CIFAR-10 dataset [\[35\]](#). The DPSPSM achieves an evasion success rate of 30.45% on the CIFAR-10 dataset [\[35\]](#) and 29.30% on the CIFAR-100 dataset [\[35\]](#).

Results and Discussion

Dataset

The CIFAR-10 dataset [\[35\]](#) comprises 60,000 color images of size 32×32 distributed equally across 10 distinct categories, amounting to 6000 images per category. This collection is partitioned into 50,000 training samples and 10,000

testing samples. Similarly, the CIFAR-100 dataset [35] shares the same structural framework as CIFAR-10, but contains 100 classes with 600 images allocated to each. For every class in this dataset, there are 500 images designated for training and 100 images for testing.

Results

DPSPSM is classified as a white-box adversarial attack. This implies that the method leverages the internal architecture and parameter values of the target network rather than treating it as an opaque black box.

The computational simulations were executed on a virtual machine provided via Google Colab. The evaluation utilized the VGG-16 [56] architecture evaluated against the test sets of both CIFAR 10 [35] and CIFAR 100 [35]. The baseline accuracy of the uncorrupted VGG-16 model reached 93.80% on the clean CIFAR 10 test set and 70.5% on the CIFAR 100 dataset. The efficacy of an adversarial method is defined as the mathematical difference between this baseline performance and the degraded accuracy recorded after deploying the respective attacks on the test data. Table 1 outlines the attack success rates for the SPSM method, the proposed DPSPSM technique, the One Pixel Attack [32], and the FGSM [13] approach when evaluated on the CIFAR 10 dataset [35]. For the CIFAR 100 dataset [35], the SPSM Attack (configured with $K = 30$) demonstrates a success rate of 35%, whereas the DPSPSM attack attains an efficiency of 29.30%. The data compiled in Table 1 indicates that while DPSPSM achieves performance metrics closely trailing the One Pixel Attack [32], it does not outperform either it or FGSM [13]; nevertheless, DPSPSM offers a substantial reduction in computational requirements compared to the one-pixel method. Furthermore, while FGSM injects perturbations across the entire image space, DPSPSM restricts its modifications to a single vertical seam.

Figure 2 illustrates the adversarial examples produced by applying the SPSM attack with parameter settings of $K = 10$ (a) and $K = 30$ (b) to images from the CIFAR 100 dataset. The visuals demonstrate that localized clusters of perturbations may emerge, occasionally obscuring the target features to such an degree that even human observers struggle to correctly distinguish the primary subject.

Table 1. Results of Attacks on CIFAR-10 [35] test data set.

Attack Name	Success Rate (%)
SPSM Attack ($K = 10$)	10
SPSM Attack ($K = 20$)	21.5
SPSM Attack ($K = 30$)	32.5
DPSPSM Attack (Proposed Attack)	30.45
One Pixel Attack [32]	31.40
FGSM [13]	90.93

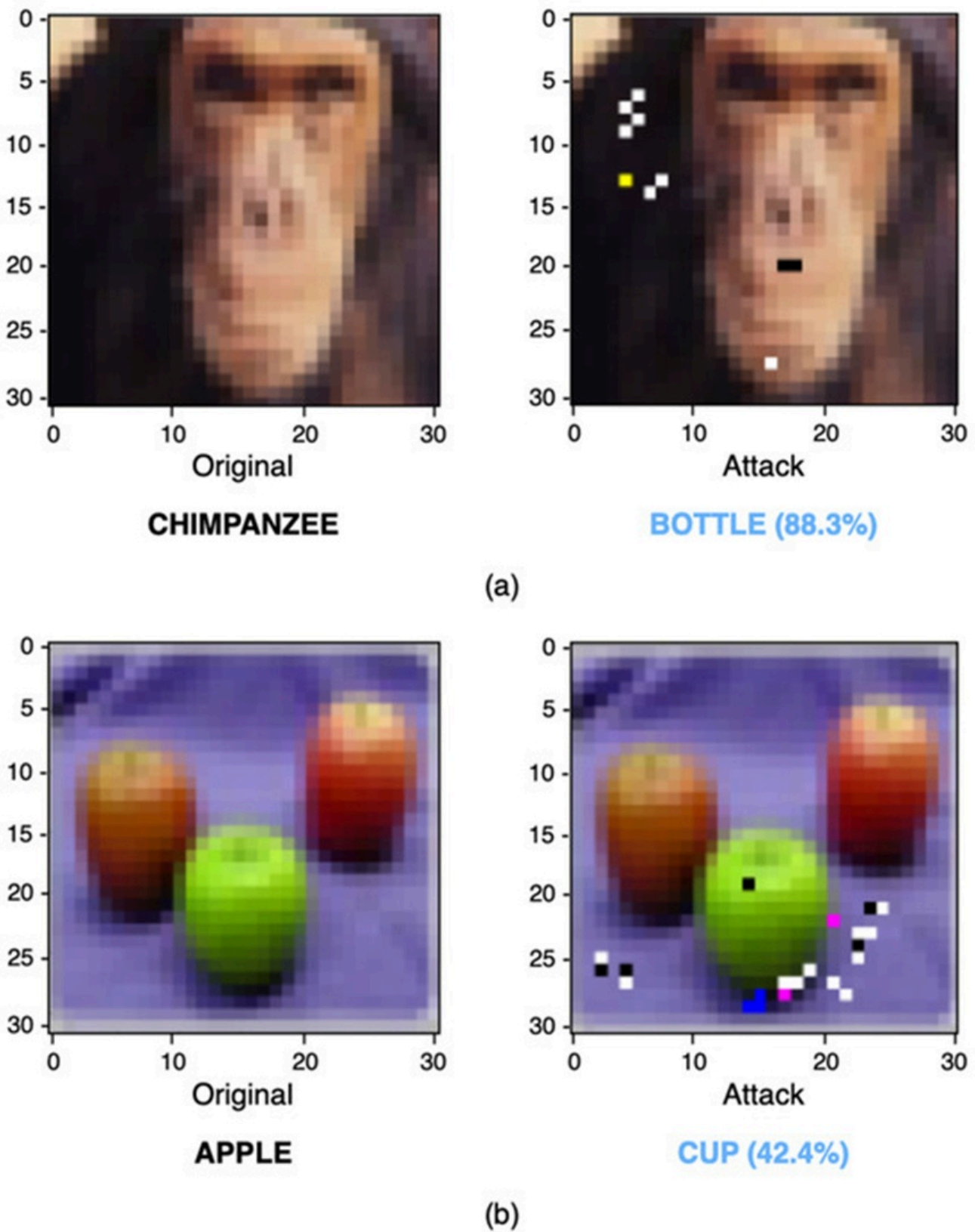
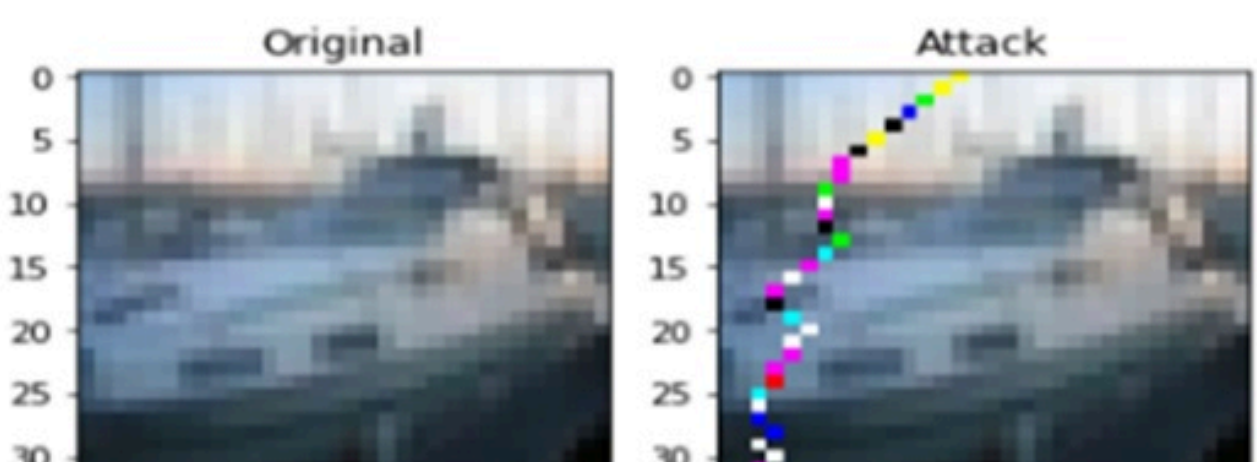
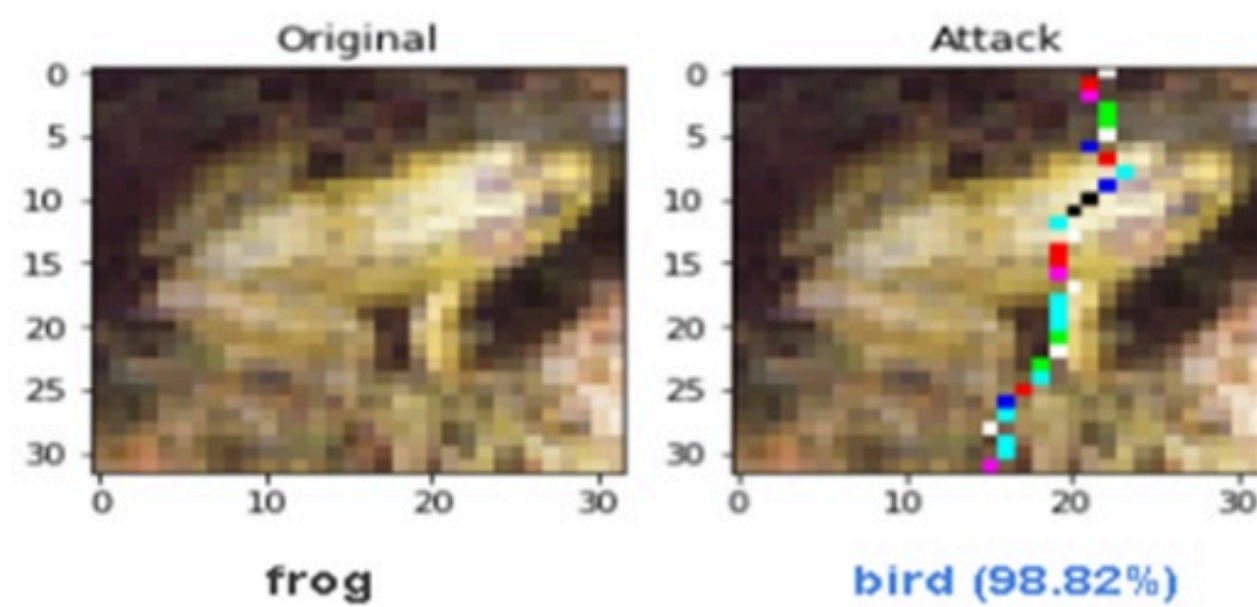
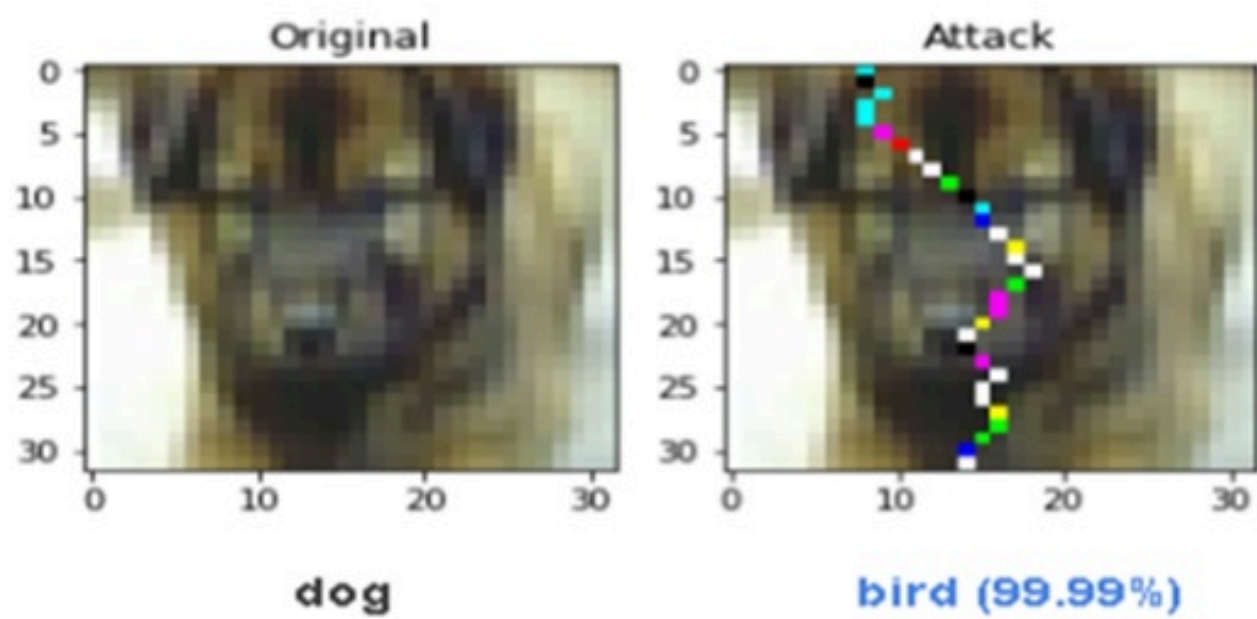
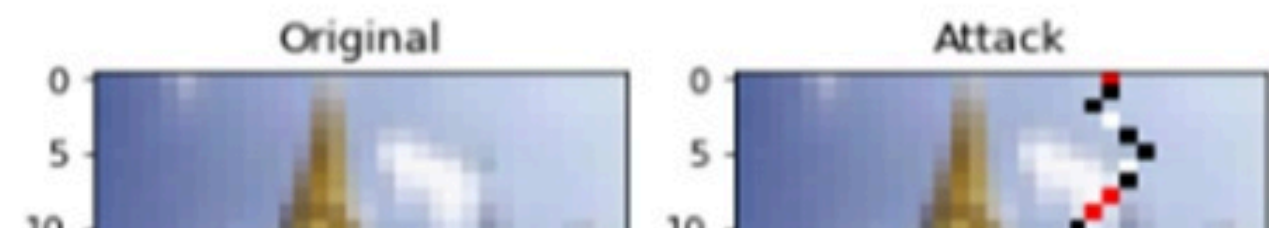
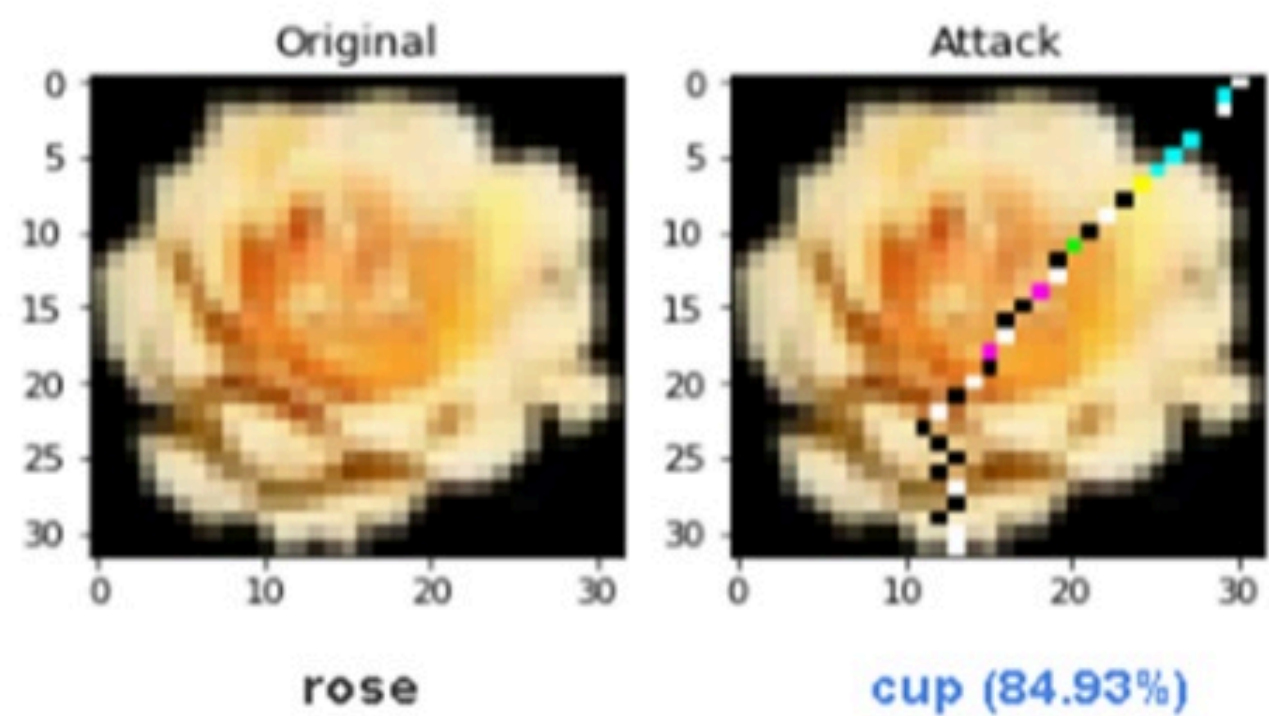
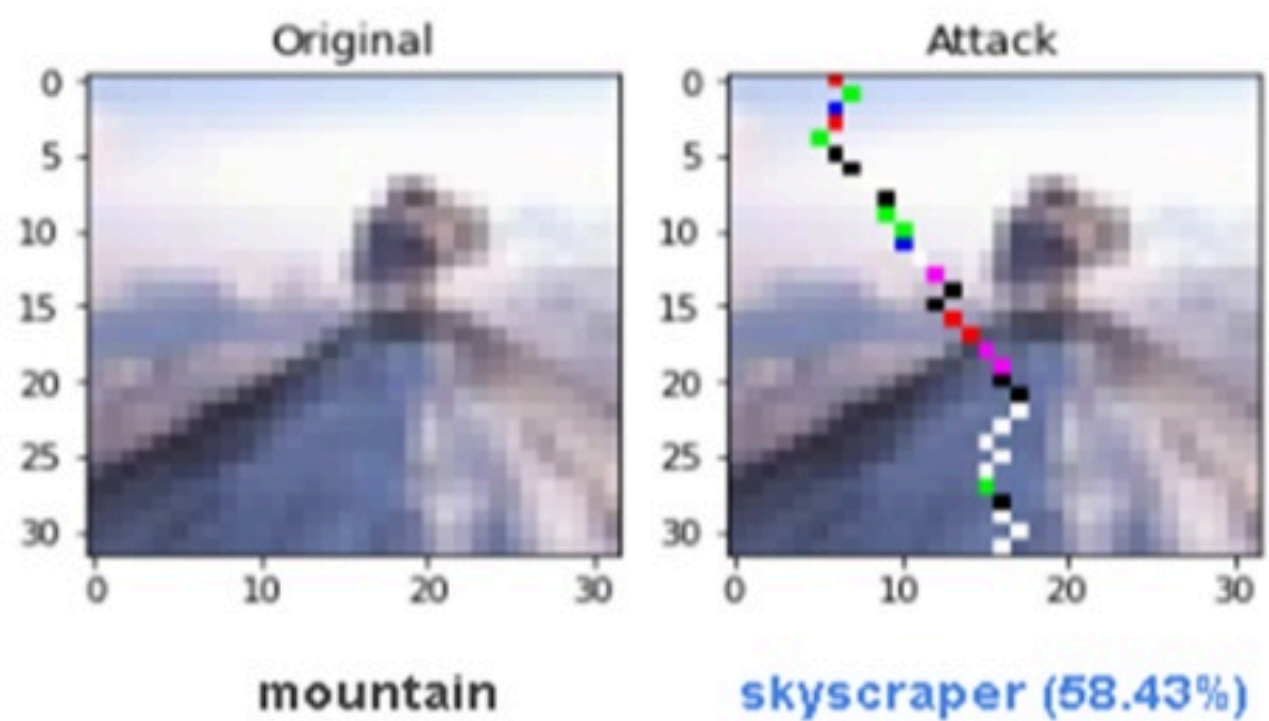


Figure 2. Examples of perturbed CIFAR-100 images generated using the SPSM attack with K = 10 (a) and K = 30 (b) along with the predicted label and corresponding confidence.

Figure 3 displays the adversarial images produced from the CIFAR-10 dataset utilizing the DPSPSM method. Meanwhile, **Figure 4** highlights the efficacy of the introduced attack scheme when applied to the CIFAR-100 dataset.







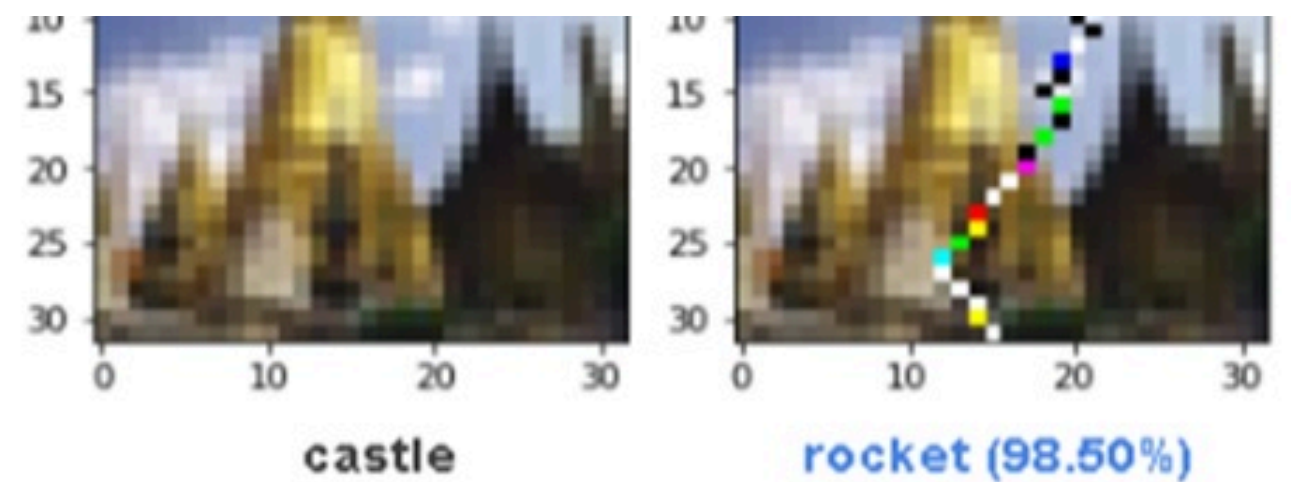


Figure 10: Perturbations regarding the adversarial vulnerability inherent to DPSPM.

The empirical findings of this analysis emphasize the narrow margin between adversarial attack potency and human visual detection. The DPSPSM method illustrates a highly strategic mechanism by isolating its perturbations to a single vertical seam. Though visually subtle, this targeted approach causes a sharp drop in model accuracy. Uncovering these low-perceptibility modifications is vital for identifying how deep neural networks fail when exposed to adversarial perturbations. The fact that human eyes effortlessly recognize the distorted images while the digital models fail underscores a fundamental divergence between biological visual processing and DNN architectures. This discrepancy exposes weaknesses in contemporary deep learning models and points toward prospective avenues of study focused on constructing networks that better replicate human visual systems, thereby bolstering their defenses against adversarial inputs.

Discussion, limitations and future work

The introduced methodology is meaningful because it establishes and explores a novel framework for disrupting DNNs, thereby prompting a reassessment of current network operations and training methodologies. While FGSM [13] effectively fulfills the demand for rapid, gradient-based attack vectors, the efficacy of both the one-pixel attack [32] and the fixed hamming method [31] reinforces the validity of utilizing localized alterations. Our experiments further validate the premise that DNN classification relies heavily on a limited subset of pixels; instead of deploying the highly potent but computationally expensive Differential Evolution (DE) algorithm characteristic of the one-pixel attack [32], DPSPSM operates by isolating and exploiting image gradients and their constituent sub-pixel values. The rapid execution of the DPSPSM algorithm coupled with these experimental successes exposes systemic vulnerabilities within these models, underscoring the susceptibility of state-of-the-art DNNs to relatively straightforward exploits.

Although the four adversarial techniques (compiled in Table 1) are all designed to deceive deep neural networks through minimal image alterations, their internal algorithms yield contrasting levels of computational efficiency, as detailed in Table 2.

Table 2. Comparative Analysis of Adversarial Attack Methods.

Aspect	SPSM	One Pixel Attack	DPSPSM	FGSM
Methodology	Sub-Pixel Score Method	Differential Evolution (DE)	Dynamic Programming (DP) combined with gradient variation	Gradient-based technique
Perturbation Strategy	Adjusts top-K pixels according to SPSM	Alters a single pixel through iterative	Changes a vertical seam using a structured	Perturbs all pixels using gradient

	scoring	optimization	optimization process	signals
Computational Overhead	Moderate, as it involves selecting and modifying top-K pixels	High, due to many iterations required by DE	Lower, since it relies on direct computation via DP and gradients	Low, because it uses a single forward pass
Efficiency	Efficient, but perturbations may be scattered, reducing overall effectiveness	Computationally expensive due to repeated iterations	Computationally efficient by directly exploiting gradient information	Highly efficient due to direct use of gradient information

The DPSPSM framework utilizes dynamic programming to isolate and alter an optimized array of pixels, targeting a singular vertical seam across the image. This strategy drastically minimizes computational latency relative to alternative methodologies that depend on extensive iterative adjustments or global image modifications. For comparison, the One Pixel Attack relies on Differential Evolution (DE) to determine the best single pixel for corruption—a process that is computationally intensive and demands numerous iterations to achieve convergence. Conversely, DPSPSM merges a gradient-driven framework with dynamic programming, facilitating a direct and more accelerated calculation.

Despite its ability to compromise an image via a single pixel modification, the One Pixel Attack demands heavy computational assets due to the cyclical optimization inherent to DE. Every individual loop necessitates evaluating the classification impact of changing various candidate pixels, a process that becomes resource-intensive for high-dimensional data matrices. In contrast, DPSPSM determines the Sub-Pixel Score Method (SPSM) metric for every pixel unit and applies dynamic programming to establish the optimal path for alteration. This procedural pipeline constrains the total number of required mathematical operations by tracking a precalculated trajectory (the seam) rather than analyzing the entire multidimensional image space.

The performance evaluations documented in [Table 1](#) of this study illustrate the comparative success rates observed on the CIFAR-10 test data. The One Pixel Attack secures a success rate of 31.40%, whereas the proposed DPSPSM reaches a comparable effectiveness of 30.45%. Notwithstanding this marginal divergence in success metrics, the computational speedup offered by DPSPSM represents a major operational benefit. The Fast Gradient Sign Method (FGSM), representing another gradient-dependent mechanism, adjusts the complete image profile to achieve a superior success rate of 90.93%, but does so by altering every pixel, which demands significantly more computational effort.

The reduced computational footprint of the DPSPSM algorithm makes it highly applicable for contexts requiring immediate adversarial generation, including real-time systems or high-throughput adversarial training pipelines. The capacity to generate rapid perturbations without suffering a major drop in success metrics offers an optimized trade-off between processing overhead and attack potency. This operational efficiency indicates that the DPSPSM framework can be effectively scaled up to accommodate larger datasets and increasingly intricate architectures without incurring unsustainable computational costs.

Our primary objective is to investigate a novel methodology for compromising these models in an accelerated yet potent manner. The observable degradation in model accuracy validates our assertion that these DNNs rely on hidden signals during their training phase; these signals frequently diverge from typical human visual perception, yielding exceptional performance on trained and homogeneous images while simultaneously leaving the models defenseless against adversarial manipulations.

The utility of the introduced framework is constrained by its classification as an untargeted white-box attack. In real-world scenarios, the underlying parameters of the target architecture are frequently inaccessible, which would render this attack ineffective since it depends on pixel gradient calculations. To mitigate this limitation and transition the framework into a black-box attack, optimization methodologies such as Bayesian optimization [\[56\]](#) could be employed to pinpoint the target pixels for perturbation. Another critical component requiring optimization is the maximum channel intensity currently

applied to the altered pixels. Modifying the channel intensity of a highly restricted subset of pixels [32] has been proven to deceive networks with high confidence metrics. Within the DPSPSM framework, channels are driven to extreme values determined by the sign of their computed gradients; nonetheless, additional empirical studies are necessary to evaluate the consequences of moving fractional distances along these gradient vectors—akin to the mechanism used in FGSM [13]—thereby adjusting the sub-pixel to an intermediate value rather than an extreme one. This highlights another critical observation: although the primary object within the image remains easily discernible and unambiguous to a human observer, the human eye can still detect the alterations due to the extreme shifts in sub-pixel intensity. We intend to modify this extreme adjustment phase by implementing an alternative technique. Rather than assigning extreme values, the target pixels will be colored based on their local surroundings using a specific mathematical function. Consequently, human observers will be unable to perceive the intentional modifications, even as the attack successfully deceives the underlying models.

Our subsequent research will focus on resolving these specified constraints and exploring methods to minimize the Hamming distance to match the conditions of the one-pixel attack [32] through a streamlined, high-speed gradient process. Future investigations could likewise expand the DPSPSM framework to alternative neural network structures, such as Recurrent Neural Networks (RNNs) and Transformer models, particularly for applications within natural language processing (NLP) and speech recognition. Furthermore, it is essential to analyze the structural resilience of DPSPSM across diverse deep neural network (DNN) topologies beyond VGG-16, including ResNet, Inception, and EfficientNet. To maximize the potency of DPSPSM, we recommend refining the Sub-Pixel Score Method (SPSM) by integrating advanced heuristics or machine learning algorithms capable of more precisely forecasting the consequences of pixel perturbations. Deploying parallel and distributed computing frameworks could also substantially minimize the runtime required to synthesize adversarial samples. Additionally, a rigorous benchmarking of DPSPSM against alternative state-of-the-art adversarial techniques across distinct datasets and problem domains should be executed to verify its scalability and generalizability. Lastly, examining hybrid methodologies that combine DPSPSM with frameworks like Generative Adversarial Networks (GANs) could facilitate the creation of highly advanced and less perceptible adversarial perturbations.

Conclusion

In this study, the authors explore a novel strategy that integrates gradient-based adversarial attacks with a constant Hamming distance, leveraging dynamic programming principles to construct the Dynamic Programming based Sub-Pixel Score Method (DPSPSM) attack vector. DPSPSM operates efficiently by capitalizing on the linearity present within the various decision boundaries of the target network. Although contemporary deep learning architectures and optimization protocols successfully deliver state-of-the-art metrics, they simultaneously introduce structural vulnerabilities to seemingly trivial input modifications. This susceptibility is demonstrated by DPSPSM, which applies minor yet highly consequential alterations to inputs, successfully misleading the networks. The presence of these vulnerabilities within highly advanced deep neural networks highlights critical security considerations and motivates further research into novel defensive mechanisms. The rapid generation pipeline and the foundational theory governing DPSPSM make it a compelling area of study. The experimental outcomes validate that the DPSPSM attack successfully misleads the target model, registering a success rate of 30.45% on the CIFAR-10 test collection and 29.30% on the CIFAR-100 test collection.

The verified potency of the DPSPSM attack in leveraging minimal yet critical perturbations emphasizes the necessity for engineered DNNs with higher robustness. Despite demonstrating exceptional accuracy on standardized benchmarks, contemporary models remain susceptible to meticulously formulated adversarial samples, revealing a fundamental divergence in how DNNs and human beings interpret visual data. To bridge this divide, subsequent studies should prioritize the engineering of models that more accurately mirror human visual processing, thereby improving their defense against adversarial interventions. This objective can be pursued by embedding more advanced heuristics or machine learning strategies that offer a superior prediction of how pixel perturbations affect classification. The DPSPSM attack not only exposes the structural liabilities of current DNN frameworks but also outlines a development pathway for more

resilient and computationally efficient adversarial attack methodologies. This research contributes to the ongoing academic dialogue surrounding adversarial machine learning and stimulates the design of innovative solutions intended to maximize the security and dependability of artificial intelligence systems. The insights derived from this analysis can steer future research toward building models that interpret images in a manner congruent with human vision, thereby upgrading their defensive resilience against such exploits and securing their safe, dependable integration into real-world settings.

Abbreviations

CNN	Convolutional Neural Network
DE	Differential Evolution
DL	Deep Learning
DNN	Deep Neural Network
DP	Dynamic Programming
DPSPSM	Dynamic Programming based Sub-Pixel Score Method Attack
FGSM	Fast Gradient Sign Method
ML	Machine Learning
SPSM	Sub-Pixel Score Method

Acknowledgements

None

Conflict of interest

None

Financial support

This research received no external funding. APC will be funded by Molde University College, Norway.

Ethics statement

None

Received: 17 Jan 2024

Revised: 08 Mar 2024

Accepted: 06 Apr 2024

Published online: 15 September 2024

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and

distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

References

- Deng L, Yu D. Deep learning: methods and applications. *Found Trends Signal Process*. 2014;7:197-387.
- Hoy MB. Alexa, Siri, Cortana, and more: an introduction to voice assistants. *Med Ref Serv Q*. 2018;37(1):81-8.
- Zhai S, Chang K-H, Zhang R, Zhang Z. DeepIntent: learning attentions for online advertising with recurrent neural networks. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; 2016 Aug 13-17; San Francisco, CA, USA. New York: ACM; 2016. p.1295-304.
<https://doi.org/10.1145/2939672.2939759>.
- Zhang H, Cao X, Ho JKL, Chow TWS. Object-level video advertising: an optimization framework. *IEEE Trans Ind Inform*. 2017;13(2):520-31.
<https://doi.org/10.1109/TII.2016.2605629>.
- Elkahky AM, Song Y, He X. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In: *Proceedings of the 24th International Conference on World Wide Web*; 2015 May 18-22; Florence, Italy. New York: ACM; 2015. p.278-88.
<https://doi.org/10.1145/2736277.2741667>.
- Cheng HT, Koc L, Harmsen J, Shaked T, Chandra T, Aradhye H, et al. Wide and deep learning for recommender systems. In: *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*; 2016 Sep 15; Boston, MA, USA. New York: ACM; 2016. p.7-10.
<https://doi.org/10.1145/2988450.2988454>.
- Wang H, Wang N, Yeung DY. Collaborative deep learning for recommender systems. In: *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; 2015 Aug 10-13; Sydney, Australia. New York: ACM; 2015. p.1235-44.
<https://doi.org/10.1145/2783258.2783273>.
- Ker J, Wang L, Rao J, Lim T. Deep learning applications in medical image analysis. *IEEE Access*. 2018;6:9375-89.
<https://doi.org/10.1109/ACCESS.2017.2788044>.
- Greenspan H, van Ginneken B, Summers RM. Guest editorial: deep learning in medical imaging: overview and future promise of an exciting new technique. *IEEE Trans Med Imaging*. 2016;35(5):1153-9.
- Suzuki K. Overview of deep learning in medical imaging. *Radiol Phys Technol*. 2017;10(3):257-73.
<https://doi.org/10.1007/s12194-017-0406-5>.
- Young T, Hazarika D, Poria S, Cambria E. Recent trends in deep learning based natural language processing. *IEEE Comput Intell Mag*. 2018;13(3):55-75.
<https://doi.org/10.1109/MCI.2018.2840738>.
- Cambria E, White B. Jumping NLP curves: a review of natural language processing research. *IEEE Comput Intell Mag*. 2014;9(2):48-57.
<https://doi.org/10.1109/MCI.2014.2307227>.
- Goodfellow I, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. *arXiv*. 2015;arXiv:1412.6572.

Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, et al. Intriguing properties of neural networks. arXiv. 2013;arXiv:1312.6199.

Yuan X, He P, Zhu Q, Li X. Adversarial examples: attacks and defenses for deep learning. *IEEE Trans Neural Netw Learn Syst*. 2019;30(9):2805-24.

Athalye A, Carlini N, Wagner D. Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. arXiv. 2018;arXiv:1802.00420.

Kurakin A, Goodfellow I, Bengio S. Adversarial examples in the physical world. arXiv. 2017;arXiv:1607.02533.

Carlini N, Wagner D. Towards evaluating the robustness of neural networks. In: *Proceedings of the 2017 IEEE Symposium on Security and Privacy*; 2017 May 22-26; San Jose, CA, USA. IEEE; 2017. p.39-57.
<https://doi.org/10.1109/SP.2017.49>.

Moosavi-Dezfooli SM, Fawzi A, Frossard P. DeepFool: a simple and accurate method to fool deep neural networks. arXiv. 2016;arXiv:1511.04599.

Wiyatno R, Xu A. Maximal Jacobian-based saliency map attack. arXiv. 2018;arXiv:1808.07945.

Moosavi-Dezfooli SM, Fawzi A, Fawzi O, Frossard P. Universal adversarial perturbations. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; 2017 Jul 21-26; Honolulu, HI, USA. IEEE; 2017. p.86-94.

Kurakin A, Goodfellow I, Bengio S. Adversarial machine learning at scale. arXiv. 2017;arXiv:1611.01236.

Liu Y, Chen X, Liu C, Song D. Delving into transferable adversarial examples and black-box attacks. arXiv. 2017;arXiv:1611.02770.

Papernot N, McDaniel P, Goodfellow I. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv. 2016;arXiv:1605.07277.

Tramèr F, Kurakin A, Papernot N, Goodfellow I, Boneh D, McDaniel P. Ensemble adversarial training: attacks and defenses. arXiv. 2020;arXiv:1705.07204.

Papernot N, McDaniel P, Wu X, Jha S, Swami A. Distillation as a defense to adversarial perturbations against deep neural networks. arXiv. 2016;arXiv:1511.04508.

Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. arXiv. 2019;arXiv:1706.06083.

Papernot N, McDaniel P, Goodfellow I, Jha S, Celik ZB, Swami A. Practical black-box attacks against machine learning. arXiv. 2017;arXiv:1602.02697.

Xu W, Evans D, Qi Y. Feature squeezing: detecting adversarial examples in deep neural networks. In: *Proceedings of the 2018 Network and Distributed System Security Symposium*; 2018 Feb 18-21; San Diego, CA, USA. Internet Society; 2018.

Zantedeschi V, Nicolae MI, Rawat A. Efficient defenses against adversarial attacks. arXiv. 2017;arXiv:1707.06728.

Shamir A, Safran I, Ronen E, Dunkelman O. A simple explanation for the existence of adversarial examples with small Hamming distance. arXiv. 2019;arXiv:1901.10861.

Su J, Vargas DV, Sakurai K. One pixel attack for fooling deep neural networks. *IEEE Trans Evol Comput*. 2019;23(5):828-41.
<https://doi.org/10.1109/TEVC.2019.2890858>.

Avidan S, Shamir A. Seam carving for content-aware image resizing. *ACM Trans Graph*. 2007;26(3):10.
<https://doi.org/10.1145/1276377.1276390>.

Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. *Commun ACM*. 2017;60(6):84-90.
<https://doi.org/10.1145/3065386>.

Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv*. 2015;arXiv:1409.1556.

Saon G, Kuo HKJ, Rennie S, Picheny M. The IBM 2015 English conversational telephone speech recognition system. In: *Proceedings of Interspeech 2015*; 2015 Sep 6-10; Dresden, Germany. 2015. p.1677-81.

Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. *arXiv*. 2014;arXiv:1409.3215.

Deng L, Li J, Huang JT, Yao K, Yu D, Seide F, et al. Recent advances in deep learning for speech research at Microsoft. In: *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*; 2013 May 26-31; Vancouver, BC, Canada. IEEE; 2013. p.8604-8.
<https://doi.org/10.1109/ICASSP.2013.6639345>.

van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, et al. WaveNet: a generative model for raw audio. *arXiv*. 2016;arXiv:1609.03499.

Zhang XL, Wu J. Deep belief networks based voice activity detection. *IEEE Trans Audio Speech Lang Process*. 2013;21(4):697-710.
<https://doi.org/10.1109/TASL.2012.2229986>.

Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification. *arXiv*. 2016;arXiv:1509.01626.

Severyn A, Moschitti A. Learning to rank short text pairs with convolutional deep neural networks. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*; 2015 Aug 9-13; Santiago, Chile. New York: ACM; 2015. p.373-82.
<https://doi.org/10.1145/2766462.2767738>.

Kong X, Yu F, Yao W, Cai S, Zhang J, Lin H. Memristor-induced hyperchaos, multiscroll and extreme multistability in fractional-order HNN: image encryption and FPGA implementation. *Neural Netw*. 2024;171:85-103.
<https://doi.org/10.1016/j.neunet.2023.12.008>.

Xiao X, Zhou X, Yang Z, Yu L, Zhang B, Liu Q, et al. A comprehensive analysis of website fingerprinting defenses on Tor. *Comput Secur*. 2024;136:103577.

Biggio B, Corona I, Maiorca D, Nelson B, Šrđić N, Laskov P, et al. Evasion attacks against machine learning at test time. In: *Advanced Information Systems Engineering*. Berlin: Springer; 2013. p.387-402.

Huang L, Joseph AD, Nelson B, Rubinstein BIP, Tygar JD. Adversarial machine learning. In: *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*; 2011 Oct 21; Chicago, IL, USA. New York: ACM; 2011. p.43-58.
<https://doi.org/10.1145/2046684.2046692>.

Eykholt K, Evtimov I, Fernandes E, Li B, Rahmati A, Xiao C, et al. Robust physical-world attacks on deep learning models. *arXiv*. 2018;arXiv:1707.08945.

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 2016 ACM SIGSAC Conference on Computer and Communications Security*; 2016 Oct 24-28; Vienna, Austria. New York: ACM; 2016. p.1528-40.
<https://doi.org/10.1145/2976749.2978392>.

Goswami G, Ratha N, Agarwal A, Singh R, Vatsa M. Unravelling robustness of deep learning based face recognition against adversarial attacks. arXiv. 2018;arXiv:1803.00401.

Athalye A, Engstrom L, Ilyas A, Kwok K. Synthesizing robust adversarial examples. arXiv. 2018;arXiv:1707.07397.

Hu W, Tan Y. Generating adversarial malware examples for black-box attacks based on GAN. arXiv. 2017;arXiv:1702.05983.

Rozsa A, Rudd EM, Boulton TE. Adversarial diversity and hard positive generation. arXiv. 2016;arXiv:1605.01775.

Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A. The limitations of deep learning in adversarial settings. arXiv. 2015;arXiv:1511.07528.

Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv. 2013;arXiv:1312.6034.

Krizhevsky A. Learning multiple layers of features from tiny images [dissertation]. Toronto: University of Toronto; 2009.

Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. arXiv. 2012;arXiv:1206.2944.