

Article

Dynamic Programming-Based White Box Adversarial Attack for Deep Neural Networks

Swati Aggarwal ^{1,*} , Anshul Mittal ² , Sanchit Aggarwal ² and Anshul Kumar Singh ²

¹ Faculty of Logistics, Molde University College, 6410 Molde, Norway

² Department of Computer Engineering, Netaji Subhas University of Technology, New Delhi 110078, India; anshulmittal0409@gmail.com (A.M.); sanchitag3698@gmail.com (S.A.); anshul98ks123@gmail.com (A.K.S.)

* Correspondence: swati.aggarwal@himolde.no

Abstract: Recent studies have exposed the vulnerabilities of deep neural networks to some carefully perturbed input data. We propose a novel untargeted white box adversarial attack, the dynamic programming-based sub-pixel score method (SPSM) attack (DPSPSM), which is a variation of the traditional gradient-based white box adversarial approach that is limited by a fixed hamming distance using a dynamic programming-based structure. It is stimulated using a pixel score metric technique, the SPSM, which is introduced in this paper. In contrast to the conventional gradient-based adversarial attacks, which alter entire images almost imperceptibly, the DPSPSM is swift and offers the robustness of manipulating only a small number of input pixels. The presented algorithm quantizes the gradient update with a score generated for each pixel, incorporating contributions from each channel. The results show that the DPSPSM deceives the model with a success rate of 30.45% in the CIFAR-10 test set and 29.30% in 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



Citation: Aggarwal, S.; Mittal, A.; Aggarwal, S.; Singh, A.K. Dynamic Programming-Based White Box Adversarial Attack for Deep Neural Networks. *AI* **2024**, *5*, 1216–1234. <https://doi.org/10.3390/ai5030059>

Academic Editor: Andrea Calimera

Received: 26 May 2024

Revised: 9 July 2024

Accepted: 10 July 2024

Published: 24 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The ever-expanding ubiquity of machine learning (ML) has reconstructed the way technology functions, integrating its functionality with day-to-day operations. Deep learning (DL) networks are no longer restricted to traditional applications of algorithmic optimizations and infrastructure designs; rather, they have steered their way into everyday life [1] in the form of virtual assistants [2], targeted online advertisements [3,4], recommendation services [5–7], medical imaging [8–10], and natural language processing [11,12]. Their generalization and expressiveness are responsible for their success but can also be attributed to their counterintuitive properties. The inner functioning of these networks is concealed in their underlying mathematical and statistical layers and hidden from human eyes, which makes it difficult to track biases and mistakes.

Recent displays of the vulnerabilities [13–16] of some carefully constructed input data in DL models raises concerns regarding the wide applicability of these models in applications in which safety is critical. While these adversarial samples produce erroneous model outputs, they continue to appear ordinary to humans. Interestingly, several attacks [13,17–21] have been presented which are proficient enough to fool the models. A few of them even work without having knowledge of any underlying architecture. Also, the same set of imperceptible input alterations, without any changes, evidently affects a wide variety of models, exposing rudimentary blind spots in our understanding of the functioning of these models [22–24]. It is said that ML models generate a boundary of separation between several output classes and use them to yield results for any input provided, thus introducing some linearity in their large decision space [13].

1.1. Motivation

Adversarial attacks constitute a substantial aspect of security and protection, because they represent a conclusive issue in artificial intelligence and ML security. Contrary to the general belief that ML models function with non-linear operations, Goodfellow et al. [13] demonstrated that it is rather the linear behavior of these models in high-dimensional space that likely causes these adversarial opportunities to exist. Many defensive mechanisms [18,25–30] have been tested, and the models have been effective to some degree, but they can be defeated by modern, more efficient attacks.

Shamir et al. [31] presented an algorithm for adversarial generation based on the concept of a fixed hamming distance. Though it is more complex and slower than the conventional gradient-based approaches, like those in [13,17], it is a novel way to manipulate only a prefixed set of input elements. This notion is further supported by the successful performance of the one-pixel attack [32] against state-of-the-art ML models, raising questions over their superior accomplishments if they are susceptible to such minor alterations.

In the domain of image manipulation, Avidan and Shamir [33] argued that the effective resizing of images requires the reckoning of geometric constraints; in addition, the image content should be assessed and factored in this calculation as well. They proposed seam carving, which supports content-aware image resizing for both reduction and expansion.

The inquisitive behavior of deep neural networks (DNNs) against adversarial attacks, like those illustrated in [13,17–21], has encouraged many to harness a swift gradient-based attack approach while concurrently restricting the expanse of infused perturbations. This hypothesis is supported by the results of [13,31,32] and is further held up by the experiments conducted as part of the current work. Our aim is to generate a simple yet effective adversarial attack that requires the manipulation of a few pixels, similar to that in [31], but one that is swift, similar to that in [13].

In this work, we present a white box adversarial attack utilizing dynamic programming (DP). First, we introduce the concept of pixel value manipulations, which further lead to the formation of the sub-pixel score method, henceforth referred to as the SPSM, which is a pixel score metric. We divulge the attack associated with the score, and later, the dynamic programming-based sub-pixel score method attack, henceforth referred to as the DPSPSM, is presented, which utilizes the SPSM with DP. The current work focuses on generating perturbations for image inputs of convolutional neural networks; future studies can be conducted to transfer our findings to other models, like text-based recurrent neural networks. The DPSPSM is swift in the generation of perturbations, unlike that in [32], but it tweaks only a few pixels, as opposed to that in [13]. The proposed attack does not beat these state-of-the-art techniques in terms of its performance metrics but combines their respective advantages.

1.2. Contributions

In this paper, we propose a novel untargeted white box adversarial attack, the dynamic programming-based sub-pixel score method (DPSPSM) attack, which is a variation of the traditional gradient-based white box adversarial approach that is limited by a fixed hamming distance using a dynamic programming-based structure. The DPSPSM attack leverages dynamic programming to identify and perturb a minimal set of pixels, specifically focusing on a single vertical seam in the image. This approach significantly reduces the computational overhead compared to that of methods that require extensive iterations or modifications across the entire image. The one-pixel attack [32], for instance, employs differential evolution (DE) to identify the optimal pixel to perturb, which is computationally intensive and requires numerous iterations to converge to a solution. In contrast, the DPSPSM utilizes a gradient-based approach combined with dynamic programming, which allows for a more direct and efficient computation. Authors in this paper propose SPSM for generating a score metric correspondence of each pixel in the image, which are further used by the proposed DPSPSM attack. The attack is a gradient based untargeted white box attack constrained in the space occupied by visual perturbations.

The novelty of the DPSPSM attack lies in its strategic targeting of a single vertical seam, which, while minimally invasive visually, significantly impairs the model's accuracy. This subtlety in alteration is crucial for understanding the vulnerabilities of DNNs to adversarial examples. The observation that humans can effortlessly recognize perturbed images, even when deep neural networks (DNNs) fail to do so, highlights a critical disparity in the visual processing mechanisms of humans and DNNs. This discrepancy not only calls into question the robustness of current DNN models but also suggests potential research directions aimed at developing models that more closely emulate human visual perception. Such advancements could significantly improve the resilience of DNNs to adversarial attacks.

The main contribution of the paper is DPSPSM, that quickly generates perturbation as opposed to [32] which is computation intensive and requires a lot of iterations. Additionally, while the FGSM [13] shows a higher success rate, it does so by modifying the entire image, which is a more detectable and less sophisticated method compared to the targeted approach of DPSPSM. This distinction is critical for developing defensive strategies against adversarial attacks, as it suggests that focusing on localized, less perceptible changes could be a more effective way to safeguard against them. The proposed attack doesn't beat these state-of-the-art techniques of FGSM [13], One Pixel Attack [32] in their success rate but blends their distinctive virtues into a novel technique which is quick and constrained at the same time. The proposed work is of interest in adversarial space as the arguments presented and corresponding trivial alterations applied under DPSPSM are vigorous in successfully manipulating model output as described in Table 1. Furthermore, the computational efficiency of DPSPSM compared to the One Pixel Attack suggests its potential for scalability and practical application in resource-constrained environments. This facet of the study enriches the current discourse on adversarial machine learning and promotes the development of more efficient and less detectable attack methodologies. Consequently, it advances the understanding of the vulnerabilities inherent in deep neural networks (DNNs).

1.3. Organization

Rest of the paper is organized as follows. Section 2 introduces the relevant literature. Section 3 establishes the methodology of proposed work describing the SPSM and DPSPSM. Section 4 reports the observed analysis. Section 5 provides a brief overview surrounding the limitations and future prospects surrounding DPSPSM. Section 6 concludes the proposed work.

2. Related Work

2.1. Adversarial Attack

ML models and DNNs have steered their way into augmenting important practical problems related to images [34,35], videos, speech [36–38], voice [39,40], text [41,42] and applications such as neural network dynamics [43], and in other applications [44]. However, not only are these models susceptible and vulnerable to various adversarial attacks [13,14,16,17,32,45–51], but the generated adversarial examples are often robust in being pertinent to several different models [23,24]. Various defensive techniques were proposed like [18,25–30] but robustness of new attacks like [18,52] rendered them futile.

2.2. Gradient Based Adversarial Approach

Goodfellow et al. [13] introduced FGSM, a gradient based attack based on the concept of linearity in high dimensional decision space. The attack requires a forward pass on the input data to compute the gradient from each involved step and incorporates a fraction of this gradient in generating the adversarial perturbations. Gradient based attacks [13,17,18,22,25,27,53], are quick in computations as they only require a forward pass over the model but tweak all the pixels in the image. DPSPSM utilizes the gradient based swift approach but also restricts the pixels changed.

2.3. Fixed Length Disturbance Based Adversarial Approach

Su et al. generated adversarial examples using Differential Evolution (DE) by modifying only one pixel [31] avoiding the problem of perceptivity measurement and expanding on the concept of fixed length disturbance [31,54,55]. The efficacy of the attack is evidenced by the challenge in detecting the generated perturbation. However, DE is computationally intensive, necessitating numerous iterations to achieve the desired outcome. In contrast, DPSPSM incorporates the principle of fixed manipulations while also leveraging the swift behavior characteristic of gradient-based attacks.

2.4. Content Aware Image Resizing

Avidan and Shamir argued that effective resizing of images should not only cling to geometric constraints. Conversely, they presented Seam carving [33], a content-aware image resizing technique for both reduction and expansion. It signifies the application of dynamic programming in the domain of image manipulation. We use this DP approach to determine the appropriate skeletal modifications in the image for DPSPSM.

2.5. Relevance

DPSPSM aims to amalgamate the robustness and celerity of gradient based approach [13] and the efficacy of predefined hamming distance [31] utilizing the DP prospects highlighted in [33]. Though, FGSM [13] is quite successful and elementary in its computations, it still adjusts all the pixels of the input image. DPSPSM aims to manipulate only a prefixed set of pixels in the input image and successfully deceive the models. DPSPSM differs from one-pixel attack [32] as their approach is based on DE, while, DPSPSM is oriented towards using Gradient based computations utilizing DP approach mentioned in [33]. Their black box approach makes them resilient, however, the computations required in DE are immense, while the gradient based process, though white box attack, involves only a simple forward pass.

DPSPSM doesn't challenge the current state-of-the-art attacks aforementioned, rather it aims to validate the austerity and lucidity of exercising modifications to only a limited number of pixels using gradient based approach combining the merits of these attacks.

3. Methodology

3.1. Introduction

Adversarial image generation can be conceptualized as an optimization task with constraints. We need to add perturbations to the original image pixels such that the actual image does not change significantly but the predicted class label now changes to an incorrect label.

There are 2 main approaches for performing the same:

1. Fixed L_0 norm (or Hamming distance)—In this approach the total number of pixels that can be perturbed are constrained with no constraint on the strength of those perturbations.
2. Fixed L_K norm—In this approach, all pixels can change with an overall constraint on the accumulated force of disturbances.

We will follow the Fixed L_0 norm approach, i.e., generation with constraint on the number of pixels that are allowed to change.

Let $J(x, \theta, y)$ be the cost function that is used while training the neural net to correctly classify images, where x is the input pixel vector (corresponding to the original input image), θ is the weights of the network that we are trying to learn and y is the ground truth output label.

Once the network has been trained, the learned weights will remain fixed while we try to change the input image pixels to cause perturbations that deviate the prediction from its original value as far as possible, ergo, generate adversarial inputs. In other words, the possibility of the distorted image belonging to the same group as the original image is to

be reduced. We are currently introducing an untargeted attack, so, we do not care about which prediction label is assigned to the perturbed image as long as it is different from its true class label.

This is equivalent to maximizing our original cost function, $J(x, \theta, y)$ with x as a parameter, instead of θ (which will now be kept constant), with a constraint on the number of pixels that can change in x as represented in (1).

Formally,

$$\begin{pmatrix} \underset{x}{\text{maximise}} J(x, \theta) \\ \text{subject to } H(x, x_0) \leq K \end{pmatrix} \quad (1)$$

where,

x_0 is the original input image that we are trying to perturb.

$H(x, x_0)$ is the Hamming distance between x and x_0 or the number of pixels in which they differ.

K is a predefined bound on the maximum number of pixels that we allow to perturb.

3.2. Perturbation Rule: 0–255 Approach

To achieve a clearer presentation, we begin by discussing the foundation of our perturbation method, which is based on a first-order Taylor expansion of the loss function. According to Goodfellow et al. [13], decision boundaries of models in high-dimensional space are piecewise linear, and this hypothesis is the basis for the existence of adversarial examples. Our objective cost function $J(x, \theta, y)$ can be approximated by only the first-order term in the Taylor series expansion, as shown in (2):

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

Since $J(x_0, \theta, y)$ (the original cost of prediction) is constant, in order to maximize $J(x, \theta, y)$, we only need to maximize $(x - x_0)^T \nabla_x J(x_0, \theta, y)$, subject to the original constraint of K pixels. This is essentially the weighted sum of sub-pixel perturbations by sub-pixel gradients.

Our perturbation method relies on the gradient, ∇ , of the cost, $J(x, \theta, y)$, with respect to the sub-pixels. By assuming that the perturbation method is based on a first-order Taylor expansion of the loss function, the monotonicity of the gradient with respect to the subpixel is automatically satisfied.

We use a simple gradient based perturbation approach. A sub-pixel is defined as the channel component of the pixel. So, if an RGB image is considered, each pixel has a R (Red), G (Green) and B (Blue) sub-pixel, expressing the corresponding color outlay. By changing either a sub-pixel or all the sub-pixels belonging to a pixel, we effectively change the actual pixel illumination.

The gradient, ∇ , gives us the direction to move in N -dimensional space (N being the number of sub-pixels in input image) for steepest increase (or decrease) in cost. The sign of the gradient with respect to each input sub-pixel tells us whether that sub-pixel value should be increased or decreased in order to increase the cost of prediction.

Since we use the fixed L_0 norm and are not constrained in the strength of perturbation for a sub-pixel, for our method, we will power it up all the way to 255, or all down all the way to 0. This will depend on the sign of the gradient, i.e., whether the gradient wants us to increase or decrease a particular sub-pixel value.

We use the notation, i_c , to refer to the sub-pixel corresponding to the c^{th} channel with respect to the i^{th} pixel.

We define the new value of the sub-pixel, $x(i_c)$, as shown in (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,

$y(i_c)$ is the final value of the i_c^{th} sub-pixel.

$x(i_c)$ is the initial value of the i_c^{th} sub-pixel.

$\nabla(i_c)$ is the gradient of the i_c^{th} sub-pixel.

$sgn(x)$ is the signum function.

The perturbation rule described in (3) will be used in DPSPSM for altering the pixel values.

3.3. Sub-Pixel Scoring Method

3.3.1. A Preliminary Score Measure for Pixel Selection

Since we can perturb at most K pixels in our attack, as depicted in (1), we need a strategy to assign a score to each sub-pixel which will help us in selecting the pixels optimally for DPSPSM, later.

A fairly simple yet reasonable strategy of scoring each sub-pixel, $D(i_c)$, is derived from our perturbation rule defined in (3) is as follows:

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

where,

$D(i_c)$ is taken as the score for the i_c^{th} sub-pixel.

$\nabla(i_c)$ is the gradient for i_c^{th} sub-pixel.

$sgn(x)$ is the signum function.

$D(i_c)$ is a score measure which signifies the maximum permissible variation authorized for i_c^{th} sub-pixel based on sign of the gradient, ∇ . Intuitively, we say that the sub-pixels which can be perturbed to a greater extent according to the gradient (and thus increasing overall cost of prediction) are more important for us.

3.3.2. Improving the Score

Adding Gradient Magnitudes

The scoring technique described in (4) only takes in the sign of the gradient and the extent to which a sub-pixel can be perturbed. It was later realized that the actual value gradient can be used as a heuristic and can give more information than is obtained by just the sign of the value.

Thus, we now refine the score, $D(i_c)$, of the i_c^{th} sub-pixel in (5).

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

Vectorising the Score

The maximum permissible perturbation value, which also served as our preliminary score measure in (4) can be combined into a single expression, $F(i_c)$, for the cases when the sign of the gradient is positive or negative (but not zero) as follows:

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

We still need to take care of two things-block pixels with zero gradient from having a non-zero resulting value from this expression, and incorporate the gradient magnitude. Both of these objectives can be achieved by the multiplication of a single term $|\nabla(i_c)|$ i.e., the absolute value of the gradient.

Thus, we finally obtain the combined expression as (7):

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

Which can be further expanded to (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 is nothing but simply the product of the gradient magnitude and the maximum permissible perturbation value for the i_c^{th} sub-pixel. Yet it is significant in terms that it can be vectorized for faster calculation of the score of each sub-pixel in the input image which is available as a matrix of sub-pixels.

Combining Sub-Pixel Scores

Since all C sub-pixels of a pixel (C being the number of channels) are not independent and are ultimately tied to the same pixel, we need to ensure that they are not selected independently. This is because at the end of our attack, the final changes will be reflected only on pixels, not individual subpixels. For i^{th} pixel, selection of either one (or all) of its C sub-pixels for DPSPSM, implies its own selection. Thus, it makes sense for our selection algorithm to work on pixels, rather than sub-pixels.

However, to be able to do so, we need to assign every pixel a score too, just like we did for every sub-pixel. We do this by taking the mean of all C sub-pixel scores for a pixel. This approach ensures that each sub-pixel contributes uniformly to the selection decision of its corresponding pixel.

Thus, for the i^{th} pixel now, we define the score $D(i)$ as (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,

$x(i_c)$ denotes the sub-pixel corresponding to i_c^{th} sub-pixel.

$\nabla(i_c)$ denotes gradient of the i_c^{th} sub-pixel.

$\text{sgn}(x)$ is the signum function.

C is the total number of channels.

We call the score generated using (9) as the SPSM score of i^{th} pixel in the image. We use this SPSM score to select pixels in the proposed DPSPSM attack.

We also evaluated adversarial attack based just on the SPSM score, SPSM attack by selecting top K pixels in descending order of their SPSM score and then modifying these K pixels to their new values using the function described in (3). The success rate corresponding to the different values of K for SPSM attack on CIFAR 10 dataset [35] is mentioned in Table 1. $K = 30$ for CIFAR 100 dataset [35] had success rate of 35%. As evident, there is a probability that the perturbations can crowd a particular patch and therefore, inhibit even the ability of humans to identify the subject in image.

3.4. Dynamic Programming Based SPSM Attack—DPSPSM

As evident from “Combining Sub-Pixel Scores”, a rudimentary attack by greedily selecting the top K pixels based on their SPSM score may result in perturbations being scattered throughout the image or even concentrated around a single region, leading to an image which even the humans cannot perceive correctly. A more structured and aligned approach is to provide a desirable configuration to the perturbations such that the visuals are not corrupted and the image semantics remain unaffected.

We utilize dynamic programming to generate the skeleton frame of the perturbations using an approach similar to the seam carving [33] method of content aware image resizing. A vertical seam is defined as the set of pixels that stretches from the top of the image to the bottom, moving left or right by at most one pixel or remaining along the same column while traversing from one row to the next. Thus, for every pixel, we consider at most the three neighboring pixels above it. The generated seam is a vertical seam in

our experiment. However, horizontal seam can also be generated in a similar way. This seam doesn't occupy much visual space across the provided image but is still capable of exploiting the vulnerabilities of the DL models in a semantically sound and structured format. The produced seam only captures a single pixel in each row.

The procedure for generating the seam is outlined as follows. Initially, the Sub-Pixel Score Matrix (SPSM) of the image is generated using Equation (9). Subsequently, DP is applied to determine the seam over the generated SPSM

The coordinates of i^{th} pixel are defined as (i_x, i_y) , where i_x is the row of the pixel and i_y is the column of the pixel. We will be using i and (i_x, i_y) interchangeably to denote the same pixel.

Therefore,

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

where,

H is height of image

W is width of the image

We now define the DP matrix, DP , and Parent matrix, P , utilized in the DPSPSM attack.

$DP(i)$ denotes the DP score of the i^{th} pixel. The value of $DP(i)$ is calculated using the recurrence relation mentioned in (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,

$D(i)$ is the SPSM score of i^{th} pixel generated using (9).

Thus, $DP(i)$ is the sum of SPSM score of the i^{th} pixel and maximum DP value of the three adjacent pixels to it in the row above it. The three adjacent pixels are those which either share an edge or a vertex with current pixel. Thus, for a pixel with row, i_x and column, i_y , the adjacent pixels will be the pixels in the row i_{x-1} and columns i_{y-1}, i_y, i_{y+1} . Since for the first row, there is now row above it, we simply equate it to SPSM score.

We further define $P(i)$, as the adjacent pixel in the row above i^{th} row which has the maximum value of DP associated to it. Therefore, $P(i)$ is calculated using (11).

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

Therefore, we use (10) and (11) to generate $DP(i)$ and $P(i)$.

After traversing all the rows, the final seam to be embraced for the perturbation is selected from the pixels of last row of the image. The pixel with the maximum $DP(i)$ is picked such that $i_x = \text{height of the image}$ and each of its sub-pixels are perturbed using the approach mentioned in (3).

Now, we traverse the row above this pixel. We select the pixel, $P(i)$ of this pixel and perturb its sub-pixels using the same approach as mentioned in (3). Now, we replace i with $P(i)$. We repeat this process till we reach the first row.

The process can be described as:

1. Select i such that $DP(i)$ is maximum and $i_x = \text{height of the image}$.
2. While $i_x > 0$:
 - a. Perturb the sub-pixels of i using (3)
 - b. $i = P(i)$

This way we generate a vertical seam spanning from the top of the image to its bottom. The same process can be configured to generate a horizontal seam.

Our approach minimizes the changes in pixels as argued by Shamir et al. [31], still using the gradient based approach of Goodfellow et al. [13] inspired by the dynamic programming seam methodology of Advin and Shamir [33].

The procedure for DPSPSM is illustrated in Figure 1 and Algorithm 1.

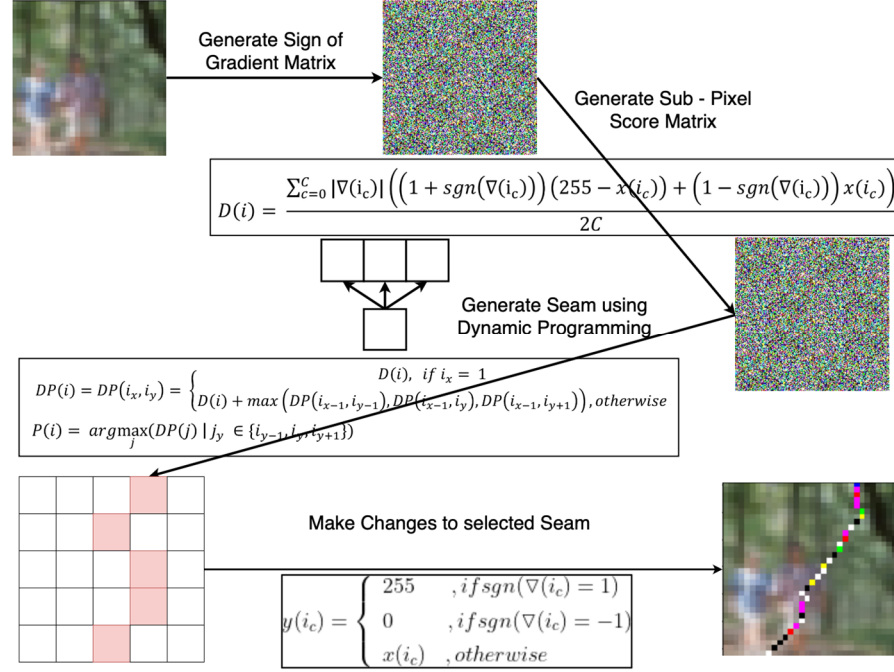


Figure 1. Flow describing the Dynamic programming based SPSM attack—DPSPSM.

Algorithm 1: Dynamic Programming Based SPSM Attack—DPSPSM

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

Output: Perturbed Images

Get_Subpixel_Score_Matrix(X, G) :

$P \leftarrow \text{Sign}(G)$

$\text{Norm}_X \leftarrow X/255$

$G \leftarrow \text{normalize}(G)$

$\text{Subpixel_Score} \leftarrow (\text{abs}(G) * ((1 + P) * (1 - G) + (1 - P) * G))/2$

$\text{Subpixel_Score_Matrix}$

$\leftarrow \text{Subpixel_Score}[0, :, :, 0] + \text{Subpixel_Score}[0, :, :, 1]$

$+ \text{Subpixel_Score}[0, :, :, 2]$

$\text{return Subpixel_Score_Matrix}$

Calculate_Max_Subpixel_Score(SSM) :

$X_{\text{matrix}} \leftarrow \text{SSM}$

$\text{Parent} \leftarrow [0 : 0]$

for row **in** $1 \dots X_{\text{matrix_shape}}[0]$:

for col **in** $1 \dots X_{\text{matrix_shape}}[1]$:

$\text{Index} \leftarrow \underset{j}{\operatorname{argmax}}(X_{\text{matrix}}[\text{row} - 1, \text{col} - 1 : \text{col} + 2])$

$\text{Parent}[\text{row}, \text{col}] \leftarrow \text{Index} + \text{col} - 1$

$X_{\text{matrix}}[\text{row}, \text{col}] += X_{\text{matrix}}[\text{row} - 1, \text{Index} + \text{col} - 1]$

$\text{return } X_{\text{matrix}}, \text{Parent}$

```

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

```

Following is the step-by-step explanation for Figure 1.

Step 1: Generate Sign of Gradient Matrix

The first step involves calculating the gradient of the cost function with respect to each sub-pixel in the image. The gradient indicates the direction in which the sub-pixel value should be adjusted to maximize the cost function.

Step 2: Generate Sub-Pixel Score Matrix

In this step, we need to generate a score matrix for each sub-pixel in the image. However, since all sub-pixels of a pixel (such as the red, green, and blue channels in an RGB image) are not independent and are ultimately tied to the same pixel, we need to ensure that they are not selected independently. This is because the final changes in our attack will be reflected on entire pixels, not just individual sub-pixels.

The score generated using (9) is called SPSM score for the i -th pixel. This SPSM score is then used to select which pixels to modify in the proposed DPSPSM

Step 3: Generate Seam Using Dynamic Programming

In this step, we generate a seam, which is a path of pixels from the top to the bottom of the image, using dynamic programming. The goal is to choose a seam that maximizes the sum of the sub-pixel scores along its path.

3.4.1. Seam Generation Process

We utilize dynamic programming to generate the skeleton frame of the perturbations, a vertical seam is defined as a set of pixels stretching from the top of the image to the bottom, moving left or right by at most one pixel or remaining in the same column while traversing from one row to the next. For every pixel, we consider at most the three neighboring pixels above it.

1. Initialization: For the first row, $DP(i)$ is simply the SPSM score of the pixel since there are no rows above it.
2. Recurrence Relation: For each subsequent row, $DP(i)$, using (10), is the sum of the SPSM score of the i -th pixel using (9) and the maximum DP value of the three adjacent pixels in the row above.
3. Parent Tracking: $P(i)$, using (11), stores the index of the pixel in the row above that has the maximum DP value.

3.4.2. Seam Selection

After traversing all the rows, the final seam is selected from the pixels in the last row of the image. The pixel with the maximum $DP(i)$ in the last row is chosen, and its sub-pixels are perturbed using (3). We then trace back the seam using the parent matrix $P(i)$:

1. Start from the Bottom: Select the pixel in the last row with the highest $DP(i)$.
2. Trace Back: Use $P(i)$ to trace back to the top row, perturbing the sub-pixels of each selected pixel along the way.

This process ensures that the generated seam spans from the top to the bottom of the image, capturing a single pixel in each row. The same method can be adapted to generate a horizontal seam if needed. This approach minimizes visual changes while effectively exploiting the vulnerabilities of deep learning models.

Step 4: Make Changes to Seam Selection

Once the seam is identified, the sub-pixels along the seam are perturbed according to the calculated scores using (3).

By following these steps, the algorithm effectively generates a structured perturbation in the form of a seam that maximizes the adversarial impact while minimizing visual distortion. Randomness in the perturbations is achieved as the specific pixels to be modified are not predetermined but are dynamically selected for each image based on the algorithm. This ensures that the perturbations are tailored to each image's unique characteristics. Additionally, the extent of pixel modification is also algorithm-dependent, allowing for adaptive perturbations that maximize the adversarial effect.

The link to the DPSPSM code is available at: <https://colab.research.google.com/drive/1W1KV3gJrGewuz5SzYyD8qg3LxuqVhsyY?usp=sharing#scrollTo=0ty48n-HhNcg> (accessed on 15 January 2024).

Table 1 mentions the success rate of DPSPSM attack on CIFAR 10 dataset [35]. DPSPSM has a success rate of 30.45% on CIFAR 10 dataset [35] and 29.30% on CIFAR 100 dataset [35].

4. Results

4.1. Dataset

The CIFAR-10 dataset [35] consists of 60,000 32×32 color images in 10 classes, with 6000 images per class. There are 50,000 training images and 10,000 test images. The CIFAR-100 dataset [35] is just like the CIFAR-10, except it has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class.

4.2. Results

DPSPSM belongs to white box attack category. i.e., this attack uses the internal structure and values of a network instead of considering it a black box.

The experiments were performed over virtual machine hosted using Google Colab. We conducted experiments on VGG-16 [56] model trained on CIFAR 10 [35] and CIFAR 100 [35] test dataset. VGG-16 model had accuracy of 93.80% on the original clean CIFAR 10 dataset and 70.5% on CIFAR 100 dataset. Success rate of attack is difference of this original accuracy and the accuracy obtained after subjecting the test set to various attacks. Table 1 signifies the success rate of SPSM attack, DPSPSM attack (the proposed attack), One Pixel Attack [32] and FGSM [13] over CIFAR 10 dataset [35]. On CIFAR 100 dataset [35], SPSM Attack (with $K = 30$) has success rate of 35% and DPSPSM attack has success rate of 29.30%. As evident from Table 1, DPSPSM performs close to the One Pixel Attack [32] but doesn't beat it or FGSM [13], however, DPSPSM has a huge computation advantage over one pixel attack. While FGSM is modifying the entire image, DPSPSM only focuses on a single vertical seam.

Figure 2 demonstrates the adversarial samples generated using the SPSM attack with $K = 10$ (a) and $K = 30$ (b) on CIFAR 100 images. As evident, there is some probability that the perturbations can crowd a particular patch and therefore, inhibit even the ability of humans to identify the subject in image.

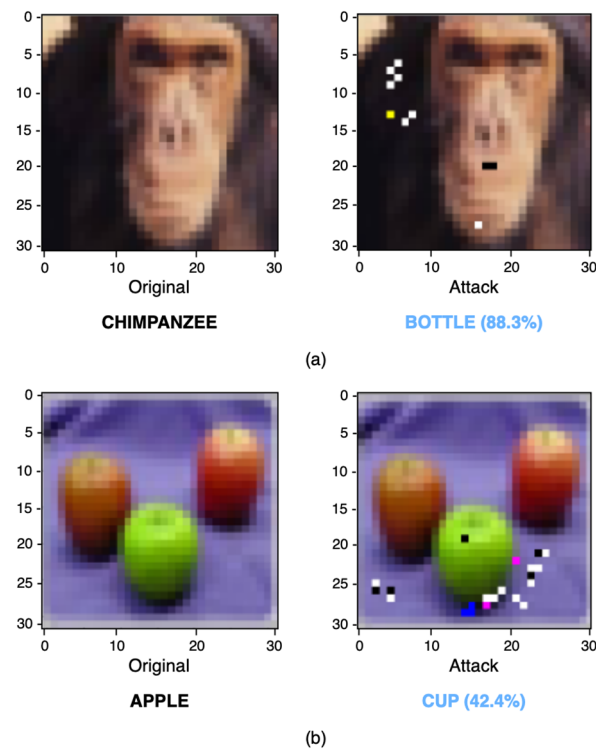


Figure 2. Examples of perturbed CIFAR-100 images generated using the SPSM attack with $K = 10$ (a) and $K = 30$ (b) along with the predicted label and corresponding confidence.

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 3 exhibits the perturbed images generated over CIFAR-10 dataset using DP-SPSM. Figure 4 demonstrates the ability of the proposed attack over CIFAR-100 dataset.

It is evident from Figures 3 and 4 that the alteration generated do not capture much visual space while at the same time are highly effective in deluding the models with a huge confidence. Humans can still easily perceive the perturbed images and identify the subject of image beyond a reasonable doubt. The image has perturbations visible to human eye, however, these perturbations are covering just a single pixel in a row and can be considered very similar to a Pole or fence in front of human eyes in real world. While the pole and fence, block some certitude of view space, the background and foreground subjects are still clearly identifiable by human eyes. Similarly, in the generated perturbed image, the modifications are introduced along a vertical seam that are not extensively blocking the subject's view, rendering it easily identifiable by humans beyond a reasonable doubt, however, the DNNs have failed to successfully recognize the image subjects.

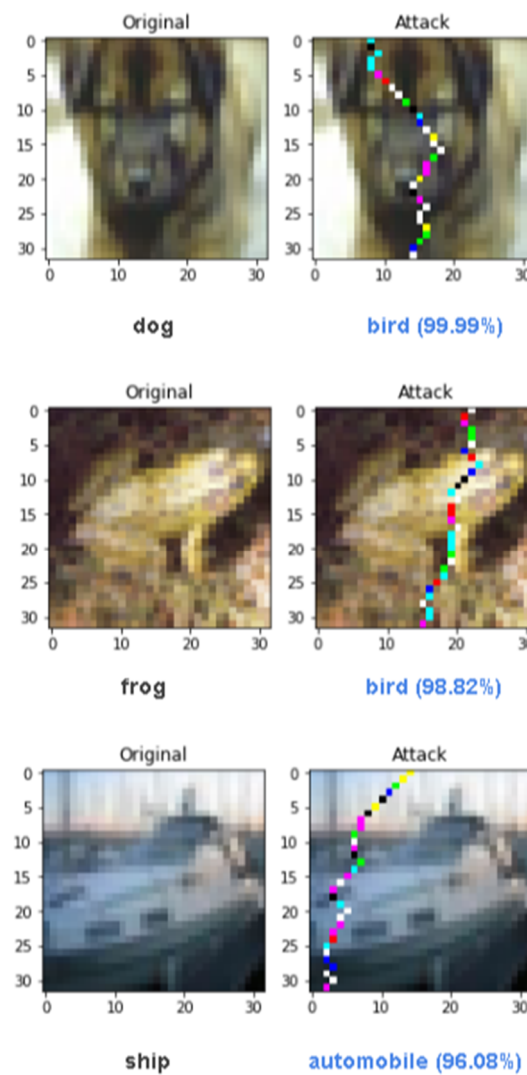


Figure 3. Examples of perturbed CIFAR-10 images generated using the proposed DPSPSM attack along with the predicted label and corresponding confidence.

The performance results tabulated in Table 1 express the robustness of the DPSPSM attack and certainly induces interest towards adversarial properties of DNN.

The results presented in this study underscore the delicate balance between the effectiveness of adversarial attacks and their perceptibility to human observers. The DPSPSM attack, in particular, showcases a strategic approach by targeting a single vertical seam. This approach, while minimally invasive visually, significantly degrades the model's accuracy. Such subtle alterations are crucial for understanding the vulnerabilities of deep neural networks (DNNs) to adversarial examples. The observation that humans can still easily recognize the perturbed images, despite the model's failure, highlights a fundamental disparity in the visual processing mechanisms of DNNs and humans. This discrepancy not only challenges the robustness of current DNN models but also suggests potential research directions aimed at developing models that more closely emulate human visual perception, thereby enhancing their resilience to such attacks.

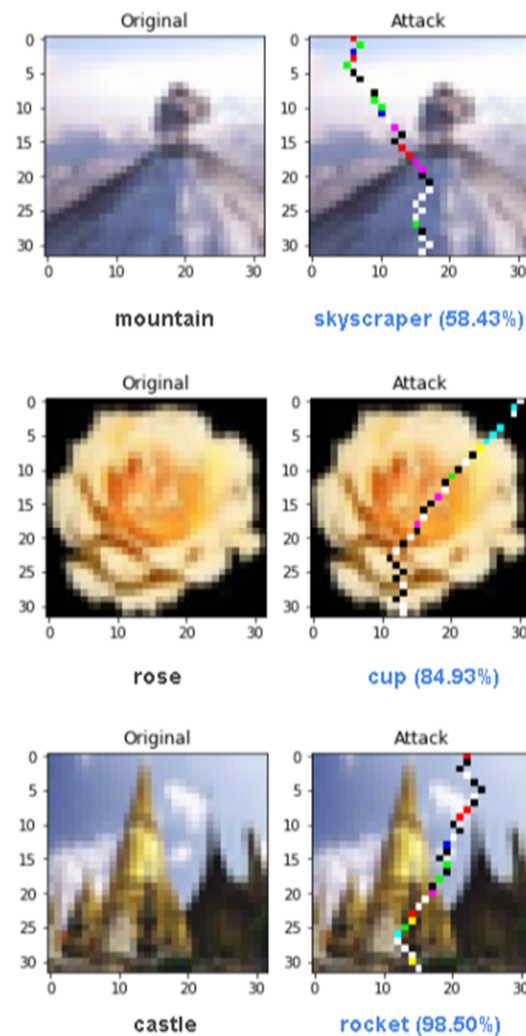


Figure 4. Examples of perturbed CIFAR-100 images generated using the proposed DPSPSM attack along with the predicted label and corresponding confidence.

5. Discussion, Limitations and Future Work

The proposed work is significant because it ploughs and unravels a new methodology of exploiting DNNs thereby, questioning the model functioning and training practices. While FGSM [13] accurately addresses the prospect of a speedy gradient based attack; the success of one-pixel attack [32] and fixed hamming approach [31] underline the notion of limited alterations. However, the certitude of DNN's dependence over only a few pixels is further ascertained by our conducted experiments as instead of using the superior yet compute demanding DE used in one-pixel attack [32], DPSPSM is solely based on exploiting the image gradient and its sub-pixel values. The swift DPSPSM process and the success of our experiment highlights the existence of numerous vulnerabilities in these models and emphasizes on the weakness of these elite DNNs to such a trivial attack.

While all four attacks (as listed in Table 1) aim to fool deep neural networks by making minimal perturbations, the underlying mechanisms lead to different computational efficiencies, as shown 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) and gradient change	Gradient-based approach
Perturbation Strategy	Modifies top-K pixels based on SPSM score	Modifies a single pixel using iterative optimization	Modifies a vertical seam using a structured approach	Modifies all pixels using gradient information
Computational Overhead	Moderate due to selection and modification of top-K pixels	High due to numerous iterations required by DE	Lower due to direct computation using DP and gradients	Low due to single forward pass
Efficiency	Efficient but may scatter perturbations that affects overall efficiency	Computationally intensive, requiring multiple iterations	Computationally efficient, leveraging gradient information directly.	Highly efficient, leveraging gradient information directly

The DPSPSM attack leverages dynamic programming to identify and perturb a minimal set of pixels, specifically focusing on a single vertical seam in the image. This approach significantly reduces the computational overhead compared to methods that require extensive iterations or modifications across the entire image. The One Pixel Attack, for instance, employs Differential Evolution (DE) to identify the optimal pixel to perturb, which is computationally intensive and requires numerous iterations to converge to a solution. In contrast, DPSPSM utilizes a gradient-based approach combined with dynamic programming, which allows for a more direct and efficient computation.

The One Pixel Attack, while effective in modifying only a single pixel, demands substantial computational resources due to the iterative nature of DE. Each iteration involves evaluating the impact of perturbing different pixels, which can be time-consuming, especially for high-dimensional images. DPSPSM, on the other hand, computes the Sub-Pixel Score Method (SPSM) score for each pixel and uses dynamic programming to select the optimal seam for perturbation. This method reduces the number of computations required, as it focuses on a predefined path (the seam) rather than evaluating the entire image space.

The experimental results presented in Table 1 of the paper demonstrate the success rates of various attacks on the CIFAR-10 dataset. While the One Pixel Attack achieves a success rate of 31.40%, DPSPSM achieves a comparable success rate of 30.45%. Despite the slight difference in success rates, the computational efficiency of DPSPSM is a significant advantage. The Fast Gradient Sign Method (FGSM), another gradient-based attack, modifies the entire image, resulting in a higher success rate of 90.93% but at the cost of altering all pixels, which is computationally more demanding.

The computational efficiency of DPSPSM makes it particularly suitable for scenarios where quick adversarial example generation is crucial, such as real-time applications or large-scale adversarial training. The ability to generate perturbations swiftly without compromising much on the success rate provides a balanced approach between computational cost and attack effectiveness. This efficiency also implies that DPSPSM can be scaled to larger datasets and more complex models without incurring prohibitive computational costs.

Our sole aim is to explore a new method of exploiting these models in a swift yet powerful manner. The perceptible decline in model’s accuracy supports our claim that these DNNs tap onto some undisclosed signals during their training which often doesn’t align with the general human perception leading to superior results over the trained and similar images but at the same time rendering these models helpless against the adversarial attacks.

The proposed work is limited in its application due to the characteristics of being an untargeted white box attack. Often the underlying model characteristics are unknown, and therefore, this attack would be inconsequential as it requires the gradient of pixels. Optimization techniques like Bayesian optimization [56] can instead be used, for determining the pixels to be perturbed, resolving this issue and yielding a black box attack. Another

key factor that requires further improvement is the channel extremity currently employed for the modified pixels. Modifying intensity of channels for even a few pixels [32] has been evident in fooling the network with great confidence value. In DPSPSM, the channels reach the extremity depending on the sign of their calculated gradient, however, more experiments are required to understand the effect of moving along a small distance in the direction of these gradient vectors like in FGSM [13], ergo, modifying the sub-pixel to some value which may or may not be extremum. This also leads to us another significant observation. Though, the object in the image is still easily identifiable and understood clearly beyond a reasonable doubt, human eyes can detect the perturbations in the image due to the extremity of change in sub-pixel value. We plan to replace this step of powering the sub-pixels to extremum with another alternative. Instead of extremum, the pixels will be colored according to its surroundings based on some mathematical function. This way the human eye won't be able to detect the deliberate change while the attack triumphs in fooling the models.

Our future efforts will be directed towards addressing the indicated shortcomings as well as exploring the perspective of minimizing hamming distance to the grounds of one-pixel attack [32] using simple swift gradient process. Future research could extend the DPSPSM approach to various other types of neural networks, including Recurrent Neural Networks (RNNs) and Transformer models, particularly in fields such as natural language processing (NLP) and speech recognition. Additionally, investigating the robustness of DPSPSM against different deep neural network (DNN) architectures beyond VGG-16, such as ResNet, Inception, and EfficientNet, is crucial. To enhance the effectiveness of DPSPSM, refining the Sub-Pixel Score Method (SPSM) to incorporate more sophisticated heuristics or machine learning techniques that can better predict the impact of pixel perturbations is recommended. Implementing parallel and distributed computing techniques could also significantly reduce the computational time required for generating adversarial examples. Comprehensive benchmarking of DPSPSM against other state-of-the-art adversarial attacks across various datasets and tasks should be conducted to evaluate the scalability and generalizability of DPSPSM. Furthermore, exploring hybrid approaches that integrate DPSPSM with techniques like Generative Adversarial Networks (GANs) could lead to the generation of more sophisticated and less detectable adversarial examples.

6. Conclusions

In this paper, the authors investigate a novel approach that combines gradient-based adversarial attacks with a fixed Hamming distance, utilizing dynamic programming to develop the Dynamic Programming based Sub-Pixel Score Method (DPSPSM) attack. DPSPSM is both efficient and exploits the linearity present in various decision spaces of the underlying network. While current deep learning models and training practices are effective in achieving state-of-the-art results, they also render these models vulnerable to seemingly trivial inputs. This vulnerability is evidenced by DPSPSM, which introduces minor yet impactful alterations to inputs, successfully deceiving the models. The existence of such vulnerabilities in advanced deep neural networks underscores the severity of security issues and stimulates further research into innovative defensive strategies. The swift generation process and the fundamental underlying theory of DPSPSM make it particularly stimulating. The results demonstrate that the DPSPSM attack deceives the model with a success rate of 30.45% on the CIFAR-10 test set and 29.30% on the CIFAR-100 test set.

The demonstrated effectiveness of the DPSPSM attack in exploiting minimal yet impactful perturbations underscores the need for more robust DNNs. Despite their high performance on standard benchmarks, current models exhibit vulnerabilities to carefully crafted adversarial examples, highlighting a fundamental gap in how DNNs and humans process visual information. To bridge this gap, future research should focus on developing models that better mimic human visual perception, thereby enhancing their resilience to adversarial attacks. This can be achieved by incorporating more sophisticated heuristics

or machine learning techniques that better predict the impact of pixel perturbations. The DPSPSM attack not only underscores the vulnerabilities of current DNNs but also provides a pathway for developing more robust and efficient adversarial attack methods. This work contributes to the ongoing discourse on adversarial machine learning and encourages the development of innovative solutions to enhance the security and reliability of AI systems. The insights gained from this study can drive future research into creating models that perceive images more like humans, thereby enhancing their resilience to such attacks and ensuring their safe and reliable deployment in real-world applications.

Author Contributions: S.A. (Swati Aggarwal): Conceptualization, Formal analysis, Methodology, Supervision, Writing—Reviewing and Editing, Revision, A.M., S.A. (Sanchit Aggarwal) and A.K.S.: Conceptualization, Formal analysis, Investigation; Methodology, Writing—original draft, All authors have read and agreed to the published version of the manuscript.

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

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not Applicable. The data sets used for experiments and analysis are public and available.

Conflicts of Interest: The authors declare no conflicts of interest.

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

References

- Deng, L.; Yu, D. Deep Learning: Methods and Applications. *Found. Trends Signal Process.* **2014**, *7*, 197–387. [\[CrossRef\]](#)
- Hoy, M.B. Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants. *Med. Ref. Serv. Q.* **2018**, *37*, 81–88. [\[CrossRef\]](#) [\[PubMed\]](#)
- Zhai, S.; Chang, K.-H.; Zhang, R.; Zhang, M. DeepIntent: Learning Attentions for Online Advertising with Recurrent Neural Networks. In KDD’16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; San Francisco, CA, USA, 13–17 August 2016. [\[CrossRef\]](#)
- Zhang, H.; Cao, X.; Ho, J.K.L.; Chow, T.W.S. Object-Level Video Advertising: An Optimization Framework. *IEEE Trans. Ind. Inform.* **2017**, *13*, 520–531. [\[CrossRef\]](#)
- Elkahky, A.M.; 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, Florence, Italy, 18–22 May 2015. [\[CrossRef\]](#)
- Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; et al. Wide & Deep Learning for Recommender Systems. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems—DLRS 2016, Boston, MA, USA, 15 September 2016. [\[CrossRef\]](#)
- Wang, H.; Wang, N.; Yeung, D.-Y. Collaborative Deep Learning for Recommender Systems. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015. [\[CrossRef\]](#)
- Ker, J.; Wang, L.; Rao, J.; Lim, T. Deep Learning Applications in Medical Image Analysis. *IEEE Access* **2018**, *6*, 9375–9389. [\[CrossRef\]](#)
- Greenspan, H.; van Ginneken, B.; Summers, R.M. Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique. *IEEE Trans. Med. Imaging* **2016**, *35*, 1153–1159. [\[CrossRef\]](#)
- Suzuki, K. Overview of deep learning in medical imaging. *Radiol. Phys. Technol.* **2017**, *10*, 257–273. [\[CrossRef\]](#)

11. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent Trends in Deep Learning Based Natural Language Processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [\[CrossRef\]](#)
12. Cambria, E.; White, B. Jumping NLP Curves: A Review of Natural Language Processing Research. *IEEE Comput. Intell. Mag.* **2014**, *9*, 48–57. [\[CrossRef\]](#)
13. Goodfellow, I.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2015**, arXiv:1412.6572. [\[CrossRef\]](#)
14. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199. [\[CrossRef\]](#)
15. Yuan, X.; He, P.; Zhu, Q.; Li, X. Adversarial Examples: Attacks and Defenses for Deep Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 2805–2824. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Athalye, A.; Carlini, N.; Wagner, D. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. *arXiv* **2018**, arXiv:1802.00420. [\[CrossRef\]](#)
17. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial examples in the physical world. *arXiv* **2017**, arXiv:1607.02533. [\[CrossRef\]](#)
18. Carlini, N.; Wagner, D. Towards Evaluating the Robustness of Neural Networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–24 May 2017; pp. 39–57. Available online: <https://ieeexplore.ieee.org/document/7958570> (accessed on 15 January 2024).
19. Moosavi-Dezfooli, S.-M.; Fawzi, A.; Frossard, P. DeepFool: A simple and accurate method to fool deep neural networks. *arXiv* **2016**, arXiv:1511.04599. [\[CrossRef\]](#)
20. Wiyatno, R.; Xu, A. Maximal Jacobian-based Saliency Map Attack. *arXiv* **2018**, arXiv:1808.07945. [\[CrossRef\]](#)
21. Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; Frossard, P. Universal Adversarial Perturbations. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 86–94. Available online: <https://arxiv.org/abs/1610.08401> (accessed on 15 January 2024).
22. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial Machine Learning at Scale. *arXiv* **2017**, arXiv:1611.01236. [\[CrossRef\]](#)
23. Liu, Y.; Chen, X.; Liu, C.; Song, D. Delving into Transferable Adversarial Examples and Black-box Attacks. *arXiv* **2017**, arXiv:1611.02770. [\[CrossRef\]](#)
24. Papernot, N.; McDaniel, P.; Goodfellow, I. Transferability in Machine Learning: From Phenomena to Black-Box Attacks using Adversarial Samples. *arXiv* **2016**, arXiv:1605.07277. [\[CrossRef\]](#)
25. Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. Ensemble Adversarial Training: Attacks and Defenses. *arXiv* **2020**, arXiv:1705.07204. [\[CrossRef\]](#)
26. 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. [\[CrossRef\]](#)
27. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv* **2019**, arXiv:1706.06083. [\[CrossRef\]](#)
28. Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z.B.; Swami, A. Practical Black-Box Attacks against Machine Learning. *arXiv* **2017**, arXiv:1602.02697. [\[CrossRef\]](#)
29. 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, San Diego, CA, USA, 18–21 February 2018. [\[CrossRef\]](#)
30. Zantedeschi, V.; Nicolae, M.-I.; Rawat, A. Efficient Defenses Against Adversarial Attacks. *arXiv* **2017**, arXiv:1707.06728. [\[CrossRef\]](#)
31. 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. [\[CrossRef\]](#)
32. Su, J.; Vargas, D.V.; Sakurai, K. One Pixel Attack for Fooling Deep Neural Networks. *IEEE Trans. Evol. Comput.* **2019**, *23*, 828–841. [\[CrossRef\]](#)
33. Avidan, S.; Shamir, A. Seam carving for content-aware image resizing. *ACM Trans. Graph. (TOG)* **2007**, *26*, 10. [\[CrossRef\]](#)
34. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2012**, *60*, 84–90. [\[CrossRef\]](#)
35. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556. [\[CrossRef\]](#)
36. Saon, G.; Kuo, H.-K.J.; Rennie, S.; Picheny, M. The IBM 2015 English Conversational Telephone Speech Recognition System. *arXiv* **2015**, arXiv:1505.05899. [\[CrossRef\]](#)
37. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. *arXiv* **2014**, arXiv:1409.3215. [\[CrossRef\]](#)
38. Deng, L.; Li, J.; Huang, J.-T.; Yao, K.; Yu, D.; Seide, F.; Seltzer, M.; Zweig, G.; He, X.; Williams, J.; 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, Vancouver, BC, Canada, 26–31 May 2013. Available online: <https://ieeexplore.ieee.org/abstract/document/6639345> (accessed on 15 January 2024).
39. van den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Kavukcuoglu, K. WaveNet: A Generative Model for Raw Audio. *arXiv* **2016**, arXiv:1609.03499. [\[CrossRef\]](#)
40. Zhang, X.-L.; Wu, J. Deep Belief Networks Based Voice Activity Detection. *IEEE Trans. Audio Speech Lang. Process.* **2013**, *21*, 697–710. [\[CrossRef\]](#)
41. Zhang, X.; Zhao, J.; LeCun, Y. Character-level Convolutional Networks for Text Classification. *arXiv* **2016**, arXiv:1509.01626. [\[CrossRef\]](#)

42. 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—SIGIR '15, Santiago, Chile, 9–13 August 2015. [\[CrossRef\]](#)
43. 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. [\[CrossRef\]](#) [\[PubMed\]](#)
44. Xiao, X.; Zhou, X.; Yang, Z.; Yu, L.; Zhang, B.; Liu, Q.; Luo, X. A comprehensive analysis of website fingerprinting defenses on Tor. *Comput. Secur.* **2024**, *136*, 103577. [\[CrossRef\]](#)
45. Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Šrndić, N.; Laskov, P.; Giacinto, G.; Roli, F. Evasion Attacks against Machine Learning at Test Time. In *Advanced Information Systems Engineering*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 387–402. [\[CrossRef\]](#)
46. Huang, L.; Joseph, A.D.; Nelson, B.; Rubinstein, B.I.P.; Tygar, J.D. Adversarial machine learning. In Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence—AISec '11, Chicago, IL, USA, 21 October 2011. [\[CrossRef\]](#)
47. Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; Song, D. Robust Physical-World Attacks on Deep Learning Models. *arXiv* **2018**, arXiv:1707.08945. [\[CrossRef\]](#)
48. Sharif, M.; Bhagavatula, S.; Bauer, L.; Reiter, M.K. 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—CCS'16, Vienna, Austria, 24–28 October 2016. [\[CrossRef\]](#)
49. 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. [\[CrossRef\]](#)
50. Athalye, A.; Engstrom, L.; Ilyas, A.; Kwok, K. Synthesizing Robust Adversarial Examples. *arXiv* **2018**, arXiv:1707.07397.
51. Hu, W.; Tan, Y. Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN. *arXiv* **2017**, arXiv:1702.05983. [\[CrossRef\]](#)
52. Rozsa, A.; Rudd, E.M.; Boulton, T.E. Adversarial Diversity and Hard Positive Generation. *arXiv* **2016**, arXiv:1605.01775. [\[CrossRef\]](#)
53. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The Limitations of Deep Learning in Adversarial Settings. *arXiv* **2015**, arXiv:1511.07528. [\[CrossRef\]](#)
54. Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv* **2013**, arXiv:1312.6034. [\[CrossRef\]](#)
55. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. Ph.D. Thesis, University of Toronto, Toronto, ON, Canada, 2009. Available online: <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf> (accessed on 15 January 2024).
56. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. *arXiv* **2012**, arXiv:1206.2944. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.