

# Adversarial attacks to image classification systems using evolutionary algorithms

Sergio Nesmachnow  
Universidad de la República  
Uruguay  
sergion@fing.edu.uy

Jamal Toutouh  
ITIS, Universidad de Málaga  
Spain  
jamal@lcc.uma.es

## ABSTRACT

Image classification currently faces significant security challenges due to adversarial attacks, which consist of intentional alterations designed to deceive classification models based on artificial intelligence. This article explores an approach to generate adversarial attacks against image classifiers using a combination of evolutionary algorithms and generative adversarial networks. The proposed approach explores the latent space of a generative adversarial network with an evolutionary algorithm to find vectors representing adversarial attacks. The approach was evaluated in two case studies corresponding to the classification of handwritten digits and object images. The results showed success rates of up to 35% for handwritten digits, and up to 75% for object images, improving over other search methods and reported results in related works. The applied method proved to be effective in handling data diversity on the target datasets, even in problem instances that presented additional challenges due to the complexity and richness of information.

## CCS CONCEPTS

• **Computing methodologies** → **Bio-inspired approaches**; *Neural networks*.

## KEYWORDS

Adversarial attacks, Generative Adversarial Networks, evolutionary algorithms, latent space search, image generation

### ACM Reference Format:

Sergio Nesmachnow and Jamal Toutouh. 2025. Adversarial attacks to image classification systems using evolutionary algorithms. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 INTRODUCTION

In the field of machine learning, image classification has emerged as a cornerstone application with significant impact across various technology fields and industries, including security, healthcare, and personalized marketing [24]. Image recognition and classification systems play a vital role in applications designed to locate objects, identify individuals, and detect features in images. However,

challenges such as improving precision and enhancing robustness have become key in advancing towards fully functional, reliable, and independent image recognition systems [23]. Among these challenges, the vulnerability of image classification methods to adversarial attacks has gained increasing attention [26].

An adversarial attack involves deliberately altering input data to mislead a machine learning model into making incorrect predictions [26]. The induced errors pose serious risks to the security and privacy of critical systems, e.g., surveillance and public safety systems [25].

This article presents an approach for generating adversarial attacks against image classifiers using a combination of evolutionary algorithms (EAs) and Generative Adversarial Networks (GANs). GANs learn data distributions to generate synthetic data that closely resembles real samples. GANs have been applied in many scientific and commercial fields, especially for generating synthetic images and videos [13]. The methodology is based on searching the latent space of GANs using EAs to find suitable vectors for generating images representing adversarial attacks against specific image classifiers. This approach advances over state-of-the-art models in literature [8] by proposing and analyzing two new fitness functions explicitly designed to optimize adversarial attacks, which balance classifier confusion and misclassification rates. Two case studies are addressed: generating attacks on classifiers for handwritten digits and object images. The proposed approach was designed to create effective (i.e., able to deceive the classifier), diverse, and high-quality adversarial examples to assess the robustness of classifiers in various problem variants. Furthermore, this method is flexible and can be applied to other datasets and classifiers, as long as a trained generative model is available to produce data samples.

The obtained results showed the effectiveness of the proposed approach in generating adversarial attacks against image classifiers, achieving competitive success rates across different problems, and significantly improving the baseline method. For handwritten digits, successful adversarial examples were generated for all classes, with the highest success rate reaching 35%. For object images, the approach performed better, achieving a peak success rate of 75%.

The article is organized as follows. Next section describes the addressed problem and methodology. The description of the proposed EA for generating adversarial attacks to classifiers is presented in Section 3. Section 4 describes the application of the evolutionary search to generate adversarial attacks against image classifiers for handwritten digits and object images. The experimental analysis and results are reported in Section 5. Finally, Section 6 presents the conclusions and formulates the main lines for future work.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*Conference'17, July 2017, Washington, DC, USA*

© 2025 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 2 ADVERSARIAL ATTACKS VIA LATENT SPACE SEARCH OF GANS

This section presents the considered problem and the methodology for generating adversarial attacks via latent space search of GANs.

### 2.1 Adversarial attacks

Adversarial attacks are subtle and intentional alterations to the input data of a machine learning model, with the purpose of deceiving the system and obtaining incorrect responses to jeopardize the reliability and accuracy of the model [12]. In image classification, adversarial attacks can manifest as images with perturbations that lead image classifiers to incorrectly identify an object or identity, or to produce ambiguous predictions where the two most likely classes are too similar, under a given threshold  $\delta$ . Such scenarios can have serious implications for the security and privacy of critical systems, for example, in surveillance and public safety systems. Continuous research in generating new adversarial attacks is essential for image classification systems to stay updated and resilient against emerging attack techniques, ensuring their security, accuracy, and reliability in changing environments [5].

GANs are artificial neural networks (ANNs) specialized in learning the distributions, features, and labels of an input dataset of real data, with the main goal of generating new synthetic data samples that follow a distribution approximated to the one of real data [11]. GANs apply adversarial training between two ANNs: a generator, which is trained to create new synthetic samples taking latent space vectors as input, and a discriminator, which learns to distinguish between real and synthetic samples while providing feedback to improve the generator. The ultimate goal of the generator is to approximate the real data distribution and produce synthetic data that is virtually indistinguishable from real data to deceive the discriminator.

The latent space of GANs is a continuous multidimensional space sampled from a random distribution (e.g., Gaussian or uniform). Traditional gradient-based search methods often struggle to navigate it effectively due to its high dimensionality and lack of clear structure for determining useful directions. Thus, EAs and other metaheuristics have been applied to guide the search and find useful vectors for the problem at hand [18]. EAs take advantage of their high versatility to deal with different GAN architectures and latent space features and their robustness to deal with changing and noisy optimization functions. The black-box optimization approach applied by EAs allows using different surrogate functions to guide the search, without relying on gradient-based operators [27].

The overall strategy considers a set of latent search vectors in the population. The evolutionary cycle apply the traditional selection and variation operators. The hybridization with the generative approach is performed on the fitness evaluation of candidate vectors: the conditional GAN is applied to generate images that are then evaluated using different classifiers. The fitness function is defined according to different metrics that allow identifying successful attacks. This way, by applying evolutionary computation, a robust global search strategy is defined for the exploration of the latent space of the considered conditional GAN to generate synthetic images that successfully attack the evaluated classifiers.

To develop an effective method for generating adversarial attacks, certain requirements needed to be established to ensure the efficiency and robustness of the process. The first requirement (R1) is to ensure that the generated adversarial attack images have a high visual quality. The second requirement (R2) is that the generated attacks should be correctly classified by the human eye. Finally, the third requirement (R3) is generating diverse attacks and ensuring that the adversarial examples have varied characteristics.

The proposed EA for generating adversarial attacks is evaluated on two image classification datasets: MNIST (Modified National Institute of Standards and Technology) database [15] and CIFAR (Canadian Institute For Advanced Research) 10 dataset [9].

### 2.2 Related work

Several recent articles have addressed the generation of adversarial attacks via the exploration of the latent space of generative models. Trajectory-based methods have been applied for attack generation in natural language processing models [16, 17], images [10, 19, 30], and other representations [6]. Approaches have applied deterministic search, e.g. via gradient descent, or random perturbations. Deterministic search is computationally efficient, but the search is highly dependable on the initial candidate solution, the method may get stuck in local optima and is not applicable to non-differentiable or discontinuous search functions. Other approaches have applied surrogate models and direct manipulation of synthetic data [22].

Population-based methods have shown improved accuracy, but many black-box approaches for adversarial attacks have relied on specific constraints or assumptions [2, 4, 6] or applied heuristic algorithms [16]. Specific approaches have been proposed to bias the search to improve the efficiency of black-box methods for generating adversarial attacks [4]. Alzantot et al. [1] applied a genetic search for black-box generation of attacks, but only exploring perturbations instead of searching the full latent space. These methods operate in the pixel space and have shown high accuracy in black-box adversarial attacks, achieving competitive results across various configurations. However, despite their performance, they exhibit important limitations: they lack semantic guidance, are often limited to untargeted attacks, and do not leverage latent representations or apply evolutionary principles in a structured search space.

The use of multiple fitness functions for attack generation has also been explored [29]. The paper reported over 60% success on CIFAR-10 using a multi-objective GA, though the absence of a generative model limited the realism and generality of the attacks.

Closed to our research, Clare and Correia [8] generated adversarial attacks via latent space exploration using a fitness function that evaluated the proximity to the distribution of real data. A second stage was needed to evaluate if the generated samples were effective attacks or not. Their method achieved 25–30% success on Fashion-MNIST and CIFAR-10, but required post-processing, which limited integration and efficiency.

In this line of work, our article contributes a compact and flexible framework for adversarial attack generation that combines semantic latent exploration with efficient evolutionary search, using simple variation operators and adaptable fitness designs. Our EA achieved competitive results, without requiring gradients, surrogate models, or post-filtering stages.

### 3 EVOLUTIONARY ALGORITHM FOR ADVERSARIAL ATTACKS TO CLASSIFIERS

This section describes the approach applying EAs for generating adversarial attacks to classifiers.

*Solution Encoding.* Each solution is encoded as a vector of floating-point numbers, representing a specific point in the latent space of the applied GAN. The dimensionality of these vectors corresponds to the input dimension required by the generator.

*Initialization.* The population is initialized using a stochastic procedure, where each value in the solution vector is sampled from a normal distribution  $\mathcal{N}(0, 1)$ . Preliminary experiments confirmed that this simple stochastic initialization provides sufficient diversity in the initial population, enabling effective exploration of the latent space. No prior knowledge of specific features or latent space directions is required to begin the evolutionary search.

*Selection.* The tournament selection operator was applied. The parameters of the tournament selection were configured to three participants and one winner. Initial experiments showed that these settings provided a correct selection pressure to guide the evolutionary search for adversarial attacks effectively.

*Recombination.* A two-point crossover operator was applied, which showed a better recombination pattern in preliminary experiments compared to a one-point crossover and arithmetic crossover.

*Mutation.* A Gaussian mutation operator was applied, with a mean  $\mu = 0$  and standard deviation  $\sigma = 1$ . This operator effectively balanced maintaining and introducing diversity in the population while minimizing disruption to the search process.

*Replacement.* A  $\mu + \lambda$  replacement strategy was applied to maintain diversity in the population, providing a proper balance between exploration and exploitation, and rapidly finding accurate solutions.

### 4 ADVERSARIAL ATTACKS TO CLASSIFIERS OF IMAGE DATASETS

This section details the application of the proposed EA to generate adversarial attacks, evaluated on two standard image classification datasets: MNIST and CIFAR-10. The datasets, generative models, classifiers, and fitness functions are described below.

#### 4.1 Datasets

The MNIST [15] dataset comprises 70,000 grayscale images of handwritten digits. It consists of 60,000 training and 10,000 test images  $28 \times 28$  pixels in size each. This dataset is widely used for benchmarking machine learning methods in classification tasks.

The CIFAR-10 dataset [9] consists of 60,000 color images, categorized into 10 classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck). CIFAR-10 consists of a total of 60,000 examples, with 50,000 images for training and 10,000 for evaluation, where each image has a size of  $32 \times 32$  pixels. This dataset is more complex than MNIST because the images are in color and exhibit more significant intra-class variability.

#### 4.2 Generative models and classifiers

The approach for generating adversarial attacks through latent space search applies a generative model to create attacks and a classifier to recognize if the generated image is an attack or not.

Conditional GANs (CGANs) were chosen for their ability to generate diverse, class-specific samples—an essential feature for targeted attacks. Among publicly available and open-source CGANs, models that produced high-quality visual samples were selected. For each dataset, the chosen CGAN generates samples that maximize their classification accuracy of state-of-the-art classifiers. Thus, a series of preliminary experiments were carried out to choose the generators. This approach ensured that the generated adversarial examples were both visually convincing and effective for testing the robustness of classifiers.

The generator used for MNIST is based on Conditional Deep Convolutional GAN by Mirza and Osindero [20]. This model uses a 100-dimensional normally distributed latent space to generate  $28 \times 28$  grayscale images of digits. In turn, the generative model for CIFAR-10 relied on Energy-based Conditional GAN [7], which has a latent space of dimension 80 and produces  $32 \times 32$  color images.

The classifiers used to evaluate the attacks and to guide the search are the publicly available ones that provided the highest classification accuracy on the training dataset in preliminary experiments. For MNIST, classifier  $c_C$  is based on a multi-layer perceptron that achieved over 99% accuracy [3]. For CIFAR-10, classifier  $c_C$  is based on a ResNet56 architecture that achieved 94.4% accuracy [28], sufficient for evaluating the generated adversarial attacks.

#### 4.3 Fitness functions

Two different fitness functions were studied for the evaluation of candidate solutions. These fitness functions require minor adjustments to accommodate dataset-specific characteristics. The fitness functions consider the following elements:

- A latent space of dimension  $d$ ,  $\mathbb{Z} \subseteq \mathbb{R}^d$
- $\mathbb{I} \subseteq \mathbb{R}^{s \times s}$  the image space (where  $s \times s$  is the image size)
- $\mathbb{K} = \{k_0, k_1, \dots, k_l\}$  the set of  $l$  class labels
- $k \in \mathbb{K}$  the target label to attack
- $g : \mathbb{Z}, \mathbb{K} \rightarrow \mathbb{I}$  the generative model
- $\mathbb{P} \subseteq [0, 1]^l$  the probabilities assigned to each label in  $\mathbb{K}$
- A classifier  $c : \mathbb{I} \rightarrow \mathbb{P}$ , where  $c_k : \mathbb{I} \rightarrow [0, 1]$  is the probability assigned by classifier  $c$  to label  $k$

Fitness function  $f_1$  addresses adversarial attack generation by maximizing the confusion (or minimizing the confidence) in the predictions. It evaluates the extent to which the classifier avoids confidently assigning high probabilities to any label for a generated sample, including the target label  $k$ . A higher value of  $f_1$  means greater confusion in the classifier, indicating that the generated sample successfully reduces the confidence across all possible labels.

$$f_1(z) = 1 - \max_{p \in \mathbb{P}} c(g(z, k)) \quad (1)$$

Fitness function  $f_2$  aims to create a scenario where the classifier is uncertain about its prediction, reducing the likelihood of correctly classifying the generated adversarial attack. It minimizes the difference between the probabilities assigned to the target predicted label  $p$  and the second most likely label, forcing the classifier to struggle between them. Besides, it minimizes the probability of the target label  $k$ . A higher  $f_2$  value reflects increased confusion and reduced confidence in the predictions of the classifier.

$$f_2(z) = 1 - \left| \max_{p \in \mathbb{P}} c(g(z, k)) - \max_{q \in \mathbb{P} \setminus \{p\}} c(g(z, k)) \right| + 1 - c_k(g(z, k)) \quad (2)$$

These functions enabled the generation of adversarial images that significantly challenged the robustness of the classifier. The specific values for  $d$ ,  $l$ , and  $c$  are 100, 10, and  $c_M$  for MNIST, respectively, and 80, 10, and  $c_C$  for CIFAR-10.

#### 4.4 Implementation

The proposed EA was implemented using Python 3.9 and the PyGAD open-source library for evolutionary and machine learning algorithms (<https://pygad.readthedocs.io/>). PyGAD provides support for building and training ANNs using EAs.

PyGAD allows customizing each step of the proposed evolutionary approach for generating adversarial attacks, enabling a simple experimentation and facilitating the incorporation of specific components such as the classifiers and the conditional GANs used for both case studies. The implementation of the proposed method for generating adversarial attacks is available in the public repository [gitlab.fing.edu.uy/sergion/ataques-adversarios-con-algoritmos-evolutivos-y-redes-generativas-antagonicas](https://github.com/sergion/ataques-adversarios-con-algoritmos-evolutivos-y-redes-generativas-antagonicas). The experimental evaluation was performed on the high-performance computing infrastructure of the National Supercomputing Center (Cluster-UY) in Uruguay [21].

### 5 EXPERIMENTAL EVALUATION

This section describes the empirical analysis of the proposed EAs for generating adversarial attacks. All images generated during the search are stored to be evaluated later.

#### 5.1 Parameters setting

For both addressed datasets, the studied parameters included the population size ( $\#P$ ), the number of generations ( $\#g$ ) used as stopping criterion, the recombination probability ( $p_R$ ), and the mutation probability ( $p_M$ ). Other parameters, including the tournament size, the values of  $\mu = 2$ , and  $\lambda = 1$ , were set in preliminary experiments.

Candidate values for studied parameters were  $\#P$  in  $\{50, 100, 200\}$ ,  $\#g$  in  $\{200, 300, 400\}$ ,  $p_R$  in  $\{0.60, 0.75, 0.90\}$ , and  $p_M$  in  $\{10^{-3}, 10^{-2}, 10^{-1}\}$ . Each parameter configuration was evaluated on 30 independent executions for the evaluated fitness functions and problem instances.

The Friedman rank statistical test was applied to analyze the distributions. The best results were computed using  $\#P = 50$ ,  $\#g = 400$ ,  $p_R = 0.75$  and  $p_M = 10^{-1}$  for MNIST. In contrast, for CIFAR-10, results of the Friedman rank statistical test confirmed that the best results were computed using the configuration  $\#P = 100$ ,  $\#g = 400$ ,  $p_R = 0.9$  and  $p_M = 10^{-1}$ , i.e., a greater population size and a higher value of  $p_R$  were needed. The higher complexity and details of the images in the CIFAR-10 dataset required a deeper exploration of the latent space than for the MNIST dataset.

#### 5.2 Fitness evolution

Figure 1 shows the evolution of the mean fitness value for  $f_1$  (top) and  $f_2$  (bottom) for both datasets.

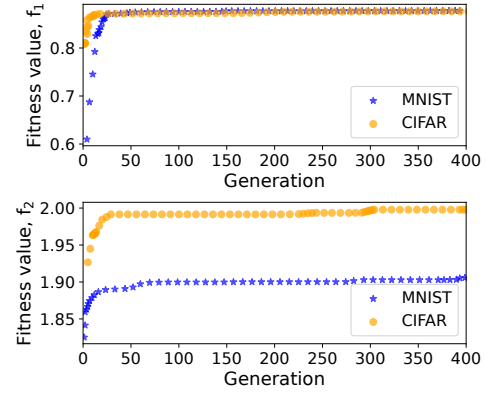


Figure 1: Mean fitness evolution for MNIST and CIFAR-10

$f_1$  increased rapidly for both datasets, rising within the first 50 generations and plateauing afterward.  $f_1$  showed similar trends on both datasets because  $f_1$  does not exploit dataset complexity or inter-class variability (CIFAR-10 is more complex and has more inter-class variability than MNIST). The goal of  $f_1$  is to maximize overall classifier confusion without targeting class similarities. In contrast,  $f_2$  exhibited distinct trends for the two datasets, emphasizing its more targeted optimization approach. For MNIST,  $f_2$  increased steadily over 150 generations, reflecting the simpler nature of this dataset and the slower process of reducing classifier confidence. For CIFAR-10,  $f_2$  rapidly increased in 50 generations, stabilizing afterward. The faster convergence shows a higher complexity of CIFAR-10, which provides more opportunities for  $f_2$  to create ambiguity between the two most probable classes. Unlike  $f_1$ ,  $f_2$  leverages dataset-specific features to generate more precise adversarial examples, leading to distinct performance differences across datasets. The smoother MNIST curves and noisier CIFAR-10 patterns suggest a more rugged fitness landscape in the latter, likely due to greater visual variability.

#### 5.3 Attacks to handwritten digits classifiers

The evaluation performed 30 independent executions of the proposed EA for each digit, using the studied fitness functions.

Table 1 reports the number of attacks generated for each digit using  $f_1$  and  $f_2$  against classifier  $c_M$ . The total number of attacks for each class are resorted in bold font. Out of 1 500 000 generated images, 24% were attacks using  $f_1$  and 35% using  $f_2$ . Digits 3, 4, and 5 were the most susceptible to adversarial attacks, with over 50 000 attacks each. In contrast, digits 6 and 9 were more challenging (fewer than 15 000 attacks). The disparity may arise from the visual similarity of digits 6 and 9 to other classes, which may complicate the generation of effective perturbations. Overall,  $f_2$  produced more attacks across most digits than  $f_1$ , but digits 6 and 9. The significantly large number of attacks found demonstrate the usefulness of the proposed approach, as this number is to be maximized.

All generated images had a high visual quality (R1) and were correctly classifiable by the human eye (R2). Regarding correctly classified images (R3, the assigned label matched the ground truth) an attack was considered successful if the two highest probabilities were within a distance  $\delta$ , indicating confusion between classes. Table 2 presents the number of generated images meeting this condition, grouped by  $\delta$  and fitness function.

**Table 1: Number of adversarial attacks generated for MNIST classifier  $c_M$ , grouped by target digit and fitness function**

fitness	target digit										total
	0	1	2	3	4	5	6	7	8	9	
$f_1$	38 098	19 624	37 742	58 512	55 661	57 647	10 409	34 411	43 550	3 918	<b>359 572</b>
$f_2$	46 386	31 555	51 813	108 399	63 309	96 490	9 622	50 141	63 860	1 698	<b>523 273</b>

**Table 2: Number of correctly classified instances in which the two highest probabilities provided by classifier  $c_M$  are within a distance less than  $\delta$ , grouped by target digit of attack and fitness function**

fitness	$\delta$	target digit										total
		0	1	2	3	4	5	6	7	8	9	
$f_1$	< 0.5	27 194	39 181	25 482	25 596	24 166	25 959	22 692	31 227	32 935	9 811	<b>264 243</b>
	< 0.4	20 927	31 333	19 773	20 553	19 968	20 722	16 959	24 418	26 161	7 022	<b>207 836</b>
	< 0.3	15 288	23 721	14 616	15 610	15 793	15 668	11 946	18 179	19 831	4 736	<b>155 388</b>
	< 0.2	10 075	16 142	9 884	10 790	11 382	10 772	7 582	12 246	13 568	2 924	<b>105 365</b>
	< 0.1	5 278	8 543	5 509	5 907	6 450	5 882	3 751	6 676	7 322	1 478	<b>56 796</b>
$f_2$	$\delta$	target digit										total
		0	1	2	3	4	5	6	7	8	9	
	< 0.5	22 593	21 311	20 236	9 617	15 687	11 252	25 616	23 886	20 771	11 679	<b>182 648</b>
	< 0.4	17 258	15 401	15 383	7 593	12 271	9 010	19 264	18 459	16 434	8 296	<b>139 369</b>
	< 0.3	12 581	10 519	11 120	5 707	9 113	6 759	13 860	13 340	12 366	5 396	<b>100 761</b>
	< 0.2	8 267	6 396	7 226	3 959	6 103	4 649	8 898	8 698	8 312	3 031	<b>65 539</b>
	< 0.1	4 320	2 978	3 739	2 119	3 196	2 494	4 444	4 371	4 349	1 255	<b>33 265</b>


More than 50 000 examples were generated where the difference between the two highest probabilities was less than 0.1 using  $f_1$ , and more than 30 000 using  $f_2$ . This finding suggests that the EA also produced samples that are correctly classified but still confuse the classifier. Although  $f_2$  was specifically designed to produce confusion between the two most probable classes, its higher overall success rate in generating misclassified attacks lowers the whole correctly produced images. Consequently,  $f_2$  produced fewer attacks of correctly classified examples compared to  $f_1$ .

Digit 3 was analyzed in detail as it exhibited the highest number of attacks. Table 3 displays the number of attacks on digit 3, grouped by fitness function, probability thresholds set to classify a prediction as an attack, and the label provided by classifier  $c_M$ .

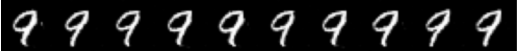
**Table 3: Number of attacks to digit 3**

fitness	$p$	class						Total
		0	2	5	7	8	9	
$f_1$	> 0	688	5 235	12 985	923	6 960	31 721	<b>58 512</b>
	> 0.5	440	4 034	8 987	548	4 344	25 985	<b>44 338</b>
	> 0.6	308	2 976	6 133	356	2 897	20 198	<b>32 868</b>
	> 0.7	178	2 092	3 907	203	1 816	15 272	<b>23 468</b>
	> 0.8	108	1 310	2 212	110	1 034	10 848	<b>15 622</b>
	> 0.9	45	635	949	43	433	6 314	<b>8 419</b>
$f_2$	$p$	class				Total		
		2	5	8	9			
	> 0	3 039	36 660	4 916	63 784	<b>108 399</b>		
	> 0.5	2 762	34 178	4 577	61 025	<b>102 542</b>		
	> 0.6	2 035	25 618	3 621	49 768	<b>81 042</b>		
	> 0.7	1 320	17 870	2 878	39 210	<b>61 278</b>		
	> 0.8	739	10 755	2 185	28 406	<b>42 085</b>		
	> 0.9	241	4 450	1 517	16 768	<b>22 976</b>		

**Table 4: Sample attacks to classifier  $c_M$  (digit 3)**

image										
class	9	9	9	9	9	9	9	9	9	9
probability	0.99	0.99	0.99	0.99	0.99	0.99	0.98	0.98	0.98	0.99

**Table 5: Sample attacks to classifier  $c_M$  (digit 9)**

image										
class	7	7	7	7	7	7	7	7	7	7
probability	0.96	0.97	0.97	0.97	0.97	0.97	0.98	0.98	0.98	0.99

A high vulnerability was evident for digit 3, as a wide variety of successful attacks were generated, even with high probability thresholds. The classifier confused digit 3 with digits 0, 2, 5, 7, 8, and 9. The attacks where the classifier assigned a probability greater than 0.9 to the incorrect digit were 8,419 when using fitness function  $f_1$  and 22,976 when using fitness function  $f_2$ . In more than 70% of these attacks, the classifier confused digit 3 with digit 9. Table 4 presents ten sample attacks generated for digit 3. Table 5 presents ten sample attacks generated for digit 9, the most challenging digit.

#### 5.4 Attacks to classifiers of common objects

Thirty independent executions of the proposed EA were performed for each object in CIFAR-10 using the studied fitness functions.

Table 6 reports the number of attacks generated for each target object using  $f_1$  and  $f_2$  against classifier  $c_C$ . All generated images had a high visual quality (R1) and were correctly classifiable by the human eye (R2). Out of all generated images, 58% were attacks against classifier  $c_C$  using  $f_1$  and 75% using  $f_2$ , significantly higher

than for MNIST. More than 150 000 attacks were generated for each object. A greater number of attacks were obtained using  $f_2$  compared to  $f_1$  for each object. This result confirms that using  $f_2$ , which explicitly focuses on generating confusion between the two most probable classes, is better to generate adversarial attacks. By promoting ambiguity and lowering the likelihood of correct predictions,  $f_2$  effectively exploits classifier vulnerabilities. Airplane was the most susceptible class to adversarial attacks, with over than 650 000 attacks. Car, horse, and deer were more challenging, with lower than 500 000 attacks.

Table 7 reports the number of correctly classified instances where the two highest probabilities had a difference less than a certain threshold  $\delta$ . More than 240 000 examples were generated with a

difference between the two highest probabilities was less than 0.1 using  $f_1$ , and more than 30 000 using  $f_2$  (similar to the results for MNIST, where  $f_1$  produced more attacks of this type than  $f_2$ ). This finding suggests that this method for generating adversarial attacks can also produce examples that are correctly classified but still confuse the classifier, mainly using fitness function  $f_1$ .

Attacks to the object airplane were analyzed in detail, as it represents the extreme in the observed distribution. More than 300 000 attacks were generated on the object airplane, suggesting it is easy to attack. Table 8 displays the number of attacks on the airplane object, organized according to the fitness function used, different probability thresholds set to classify a prediction as an attack, and the label provided by the classifier  $c_C$ .

**Table 6: Number of adversarial attacks generated for CIFAR-10 classifier  $c_C$ , grouped by target digit and fitness function**

	<i>target object</i>										<b>Total</b>
	airplane	car	bird	cat	deer	dog	frog	horse	ship	truck	
$f_1$	305 578	247 784	216 923	219 213	195 167	268 045	285 000	176 786	251 710	155 062	<b>2 321 268</b>
$f_2$	358 333	314 822	274 273	288 058	273 521	343 726	292 980	257 261	344 986	242 424	<b>2 990 348</b>


**Table 7: Number of correctly classified instances in the CIFAR-10 dataset in which the two highest probabilities are within a distance less than  $\delta$ , grouped by target object and fitness function.**

<i>fitness</i>	$\delta$	<i>target object</i>										<b>Total</b>
		airplane	car	bird	cat	deer	dog	frog	horse	ship	truck	
$f_1$	< 0.5	40 349	43 749	100 378	53 517	48 118	38 989	49 264	39 930	54 978	63 024	<b>532 296</b>
	< 0.4	36 139	37 978	97 170	45 548	41 121	33 714	45 936	33 347	47 744	52 232	<b>470 929</b>
	< 0.3	31 548	31 893	93 575	37 746	34 092	28 352	42 625	26 599	40 061	41 498	<b>407 989</b>
	< 0.2	26 003	25 113	88 941	29 362	26 295	22 605	38 943	19 535	31 256	30 033	<b>338 086</b>
	< 0.1	18 557	16 270	78 505	19 370	16 898	15 563	33 518	11 436	19 956	17 561	<b>247 634</b>
<i>fitness</i>	$\delta$	<i>target object</i>										<b>Total</b>
		airplane	car	bird	cat	deer	dog	frog	horse	ship	truck	
$f_2$	< 0.5	10 665	15 233	20 096	21 385	18 522	10 811	16 064	18 874	13 061	31 853	<b>176 564</b>
	< 0.4	8 673	12 331	16 232	17 000	14 823	8 692	12 771	14 920	10 517	25 351	<b>141 310</b>
	< 0.3	6 672	9 459	12 614	12 761	11 223	6 539	9 668	11 131	8 093	19 160	<b>107 320</b>
	< 0.2	4 591	6 512	8 746	8 574	7 696	4 446	6 557	7 499	5 512	12 956	<b>73 089</b>
	< 0.1	2 412	3 468	4 714	4 356	3 994	2 326	3 500	3 823	2 846	6 714	<b>38 153</b>


**Table 8: Number of attacks to airplane object in the CIFAR-10 dataset**

<i>fitness</i>	$p$	<i>class</i>									<b>Total</b>
		car	bird	cat	deer	dog	frog	horse	ship	truck	
$f_1$	> 0	4 835	97 149	93 824	25 691	4 933	33 536	13 242	24 494	7 874	<b>305 578</b>
	> 0.5	885	24 973	26 033	13 683	696	9 716	2 825	3 815	669	<b>83 295</b>
	> 0.6	590	20 738	19 376	10 971	519	7 779	1 762	2 908	465	<b>65 108</b>
	> 0.7	406	17 315	14 521	8 637	376	6 104	1 079	2 158	334	<b>50 930</b>
	> 0.8	268	14 050	10 538	6 502	259	4 663	635	1 473	232	<b>38 620</b>
	> 0.9	135	10 333	6 863	4 183	163	3 125	276	867	132	<b>26 077</b>
<i>fitness</i>	$p$	<i>class</i>									<b>Total</b>
		car	bird	cat	deer	dog	frog	horse	ship	truck	
$f_2$	> 0	397	88 872	112 568	52 009	4 037	44 359	32 935	20 904	2 252	<b>358 333</b>
	> 0.5	250	66 615	84 345	40 044	2 240	31 573	23 107	16 021	1 549	<b>265 744</b>
	> 0.6	186	57 024	70 743	33 546	1 816	27 122	17 501	13 467	1 159	<b>222 564</b>
	> 0.7	131	48 404	58 312	27 643	1 443	23 219	12 746	11 083	851	<b>183 832</b>
	> 0.8	80	39 881	46 036	21 974	1 128	19 061	8 677	8 734	579	<b>146 150</b>
	> 0.9	39	29 776	32 193	15 572	732	14 041	4 789	6 014	335	<b>103 491</b>

**Table 9: Sample attacks to classifier  $c_C$  with the class airplane**

image										
class	car	bird	bird	cat	deer	dog	frog	horse	ship	truck
probability	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99

**Table 10: Sample attacks to classifier  $c_C$  with the class truck**

image										
class	plane	car	bird	bird	cat	deer	dog	frog	horse	ship
probability	0.99	0.99	0.93	0.99	0.99	0.99	0.97	0.88	0.99	0.99

The classifier confused the images of airplanes with all objects in the CIFAR-10 dataset. For the airplane, 26 077 attacks were obtained where the highest assigned probability was greater than 0.9 when using fitness function  $f_1$ , and 103 491 attacks when using fitness function  $f_2$ . In both cases, the classifier confused the generated airplane with a bird or a cat in over 60% of the attacks. Table 9 presents ten sample attacks generated for the airplane class. Table 10 presents ten sample attacks generated for the truck class.

The results obtained in the evaluation of CIFAR-10 were highly positive, since multiple attacks were successfully generated for all objects in the dataset. In turn, the findings suggest that image classifiers for objects are more vulnerable than classifiers for handwritten digits. This difference could be attributed to the fact that object image classifiers require higher resolution in the processed images to distinguish details accurately.

## 5.5 Comparison with a multistart local search

The proposed EA was compared with a multistart iterated local search (MILS) method based on Alzantot et al. [1], but extended to explore the whole latent space and only perturbations.

MILS applies the Gaussian mutation operator for exploring in the neighborhood of the current solution, using  $\mu = 0$  and  $\sigma = 1$ . A predefined effort stopping criterion is applied, performing the same number of function evaluations as the proposed EA. To avoid getting stuck in a local optima, a reinitialization is applied if no improvement is found in 1000 evaluations [14].

Tables 11 and 12 report the number of attacks generated by the compared search methods for MNIST and CIFAR-10 datasets using function  $f_2$ , which allowed to compute the larger number of attacks in the experiments performed. Results show that the proposed EA generated significantly more attacks than MILS. Improvements ranged from 8.01% (digit 1) to 61.43% (digit 6) for MNIST and from 22.49% (dog) to 35.58% (car) for CIFAR-10.

Figures 2 and 3 present the boxplot comparison between EA and MILS for each class. The proposed EA improved over MILS for all classes in both case studies. Boxplots indicate that MNIST is easier to solve than CIFAR-10, as MILS computed similar results than the proposed EA for digits 1 and 6, where the improvements of EA were smaller than the inter-quartile range of the results distributions.

**Table 11: EA vs. MILS: Number of attacks to classifier  $c_M$  (MNIST) using  $f_2$** 

Target digit	Attacks (EA/MILS)	EA over MILS
0	46 386/35 003	24.54%
1	31 555/29 029	8.01%
2	51 813/37 481	27.66%
3	108 399/88 217	18.62%
4	63 309/50 315	20.52%
5	96 490/80 047	17.04%
6	9 622/3 711	61.43%
7	50 141/37 206	25.80%
8	63 860/48 288	24.38%
9	1 698/720	57.60%
<b>Total</b>	<b>523 273/410 017</b>	<b>21.64%</b>

**Table 12: EA vs. MILS: Number of attacks to classifier  $c_C$  (CIFAR-10) using  $f_2$** 

Target object	Attacks (EA/MILS)	EA over MILS
airplane	358 333/260 121	27.41%
car	314 822/202 803	35.58%
bird	274 273/183 390	33.14%
cat	288 058/204 501	29.01%
deer	273 521/209 691	23.34%
dog	343 726/266 413	22.49%
frog	292 980/227 048	22.50%
horse	257 261/198 503	22.84%
ship	344 986/261 582	24.18%
truck	242 424/179 413	25.99%
<b>Total</b>	<b>2 990 348/2 193 465</b>	<b>26.65%</b>

Other digits, such as 3 and 5, were easier to attack using EA, and significant improvements over MILS are reported. Regarding CIFAR, the specific features of images on the dataset made it harder for a simple local search method to find attacks. The improvements of the proposed EA over MILS were statistically significant for all classes. The higher improvements were computed for bird, whereas deer and truck had the smaller improvements of EA over MILS.

Figure 4 presents representative graphics of the average evolution of fitness function  $f_2$  for EA and MILS on MNIST. The proposed EA showed rapid convergence, reaching a fitness value of



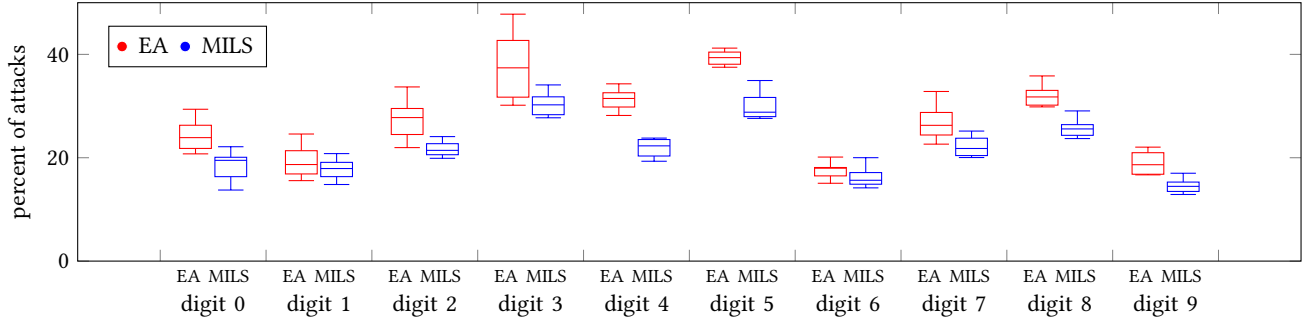


Figure 2: Comparison of attack success rates: EA vs. MILS on MNIST

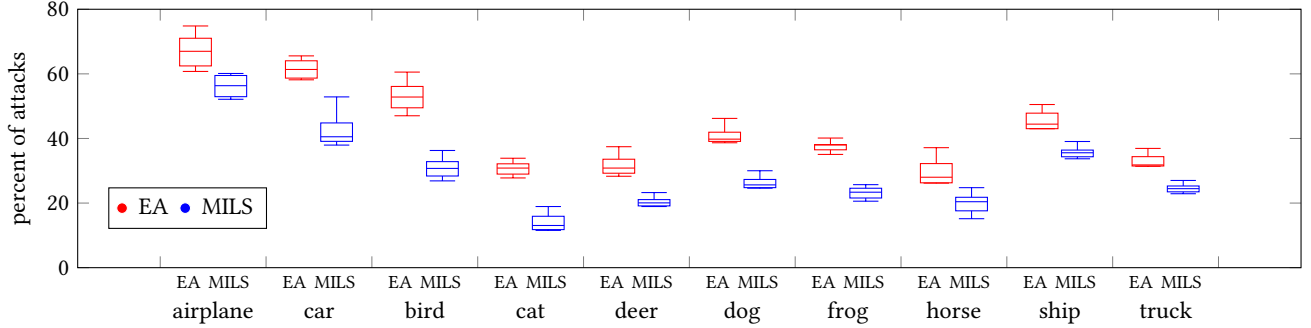


Figure 3: Comparison of attack success rates: EA vs. MILS on CIFAR-10

0.87 within 50 generations, whereas MILS had a slower progression, eventually plateauing at a lower value of approximately 0.79. The fitness evolution patterns highlight the ability of the EA to explore the latent space and maximize classifier confusion, achieving faster convergence speed and better final fitness value.

Regarding the comparison with results from the related literature, the proposed EA was competitive with results for similar problems. Wu et al. [29] reported success rates above 60% on CIFAR-10 using a multi-objective GA in the perturbation space, while Clare and Correia [8] achieved 25–30% using a two-stage latent space approach. Our method reached up to 75% success on CIFAR-10 and 35% on MNIST, using a simpler, fully integrated evolutionary framework with no gradient, surrogate model, or post-processing requirements.

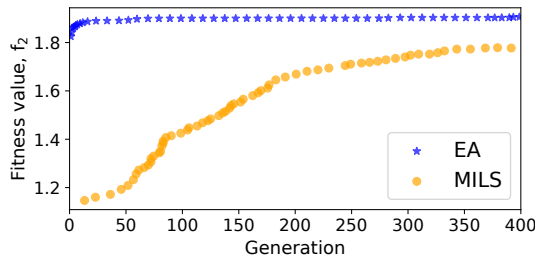


Figure 4: Fitness evolution: EA vs. MILS

## 6 CONCLUSIONS AND FUTURE WORK

This article presented an approach to generating adversarial attacks on image classifiers by leveraging the latent space of GANs through EAs, addressing a critical challenge in improving the robustness of

recognition and classification systems. The evolutionary search employed a black-box approach guided by two novel fitness functions designed to balance classifier confusion and misclassification.

Experimental results on MNIST and CIFAR-10 datasets demonstrated the effectiveness of the proposed approach, achieving success rates up to 75%, a remarkable success rate compared to related works, and significantly outperforming a MILS method. The EA showed a more consistent and fast evolution pattern for both studied fitness functions. The fitness function considering confusion between the two most probable labels allowed to generate more attacks than the one considering confidence across all possible labels. The findings revealed that object classifiers, such as those trained on CIFAR-10, are more vulnerable to adversarial attacks than simpler classifiers like those used for handwritten digits.

This article contributes to the field by demonstrating the potential of integrating EAs with GAN-based latent space exploration, offering a flexible framework for testing classifier resilience. EAs leverage their adaptability to work with various GAN architectures and latent space characteristics, along with their resilience in handling partial information thanks to the black-box optimization they applied. These features define a proper exploration pattern that allows improving other traditional methods. The adaptability of the approach allows its application to other datasets and classifier architectures, providing a valuable tool for enhancing the robustness of machine learning systems. The versatility of EAs makes the approach applicable to other generation and classification problems, including text, audio, and natural language.

The main lines for future work are related to extending the experimental validation of the proposed approach and designing more powerful variation operators. We also propose applying the methodology to the human faces recognition problem.



## REFERENCES

- [1] M. Alzantot, Y. Sharma, S. Chakraborty, H. Zhang, C. Hsieh, and M. Srivastava. GenAttack: practical black-box attacks with gradient-free optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1111–1119, 2019.
- [2] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *Computer Vision*, pages 484–501, 2020.
- [3] A. Bose. Handwritten Digit Recognition Using PyTorch: Intro to Neural Networks. <https://towardsdatascience.com/handwritten-digit-mnist-pytorch-977b5338e627>, 2019. [2024-09-08].
- [4] T. Brunner, F. Diehl, Michael T. Le, and A. Knoll. Guessing smart: Biased sampling for efficient black-box adversarial attacks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.
- [5] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay. A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology*, 6(1):25–45, 2021.
- [6] H. Chang, Y. Rong, T. Xu, W. Huang, H. Zhang, P. Cui, X. Wang, W. Zhu, and J. Huang. Adversarial attack framework on graph embedding models with limited knowledge. *IEEE Transactions on Knowledge and Data Engineering*, pages 4499–4513, 2022.
- [7] S. Chen, C. Li, and H. Lin. A Unified View of cGANs with and without classifiers. In *Advances in Neural Information Processing Systems: Proceedings of the 2021 Conference*, pages 27566–27579, 2021.
- [8] L. Clare and J. Correia. Generating Adversarial Examples through Latent Space Exploration of Generative Adversarial Networks. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, pages 1760–1767, 2023.
- [9] L. Darlow, E. Crowley, A. Antoniou, and A. Storkey. CINIC-10 Is Not ImageNet or CIFAR-10, 2018. [20-11-2024].
- [10] M. Fan, Y. Liu, C. Chen, and X. Liu. Semiadv: Query-efficient black-box adversarial attack with unlabeled images, 2024.
- [11] D. Foster. *Generative Deep Learning*. O'Reilly Media, Inc., 2019.
- [12] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations*, pages 1–9, 2015.
- [13] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye. A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3313–3332, 2023.
- [14] S. Iturriaga, S. Nesmachnow, F. Luna, and E. Alba. A parallel local search in cpu/gpu for scheduling independent tasks on large heterogeneous computing systems. *The Journal of Supercomputing*, 71(2):648–672, October 2014.
- [15] D. Li. The MNIST Database of Handwritten Digit Images for Machine Learning Research. *IEEE Signal Processing Magazine*, pages 141–142, 2012.
- [16] G. Li, B. Shi, Z. Liu, D. Kong, Yulei Wu, Xiaodan Zhang, Longtao Huang, and Honglei Lyu. Adversarial text generation by search and learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15722–15738. Association for Computational Linguistics, 2023.
- [17] S. Liu, N. Lu, C. Chen, and K. Tang. Efficient combinatorial optimization for word-level adversarial textual attack. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:98–111, 2022.
- [18] B. Machin, S. Nesmachnow, and J. Toutouh. Multi-target evolutionary latent space search of a generative adversarial network for human face generation. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2022.
- [19] A. Meißner, A. Fröhlich, and M. Geierhos. Keep it simple: Evaluating local search-based latent space editing. *SN Computer Science*, 4(6):820, 2023.
- [20] M. Mirza and S. Osindero. Conditional Generative Adversarial Nets, 2014.
- [21] Sergio Nesmachnow and Santiago Iturriaga. Cluster-UY: Collaborative Scientific High Performance Computing in Uruguay. In Springer, editor, *Supercomputing*, volume 1151 of *Communications in Computer and Information Science*, pages 188–202. Springer, 2019.
- [22] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. Celik, and A. Swami. Practical Black-Box Attacks against Machine Learning. In *Proceedings of the ACM on Conference on Computer and Communications Security*, pages 506–519, 2017.
- [23] R. Singh, A. Agarwal, M. Singh, S. Nagpal, and M. Vatsa. On the robustness of face recognition algorithms against attacks and bias. In *34<sup>th</sup> AAAI Conference on Artificial Intelligence*, pages 13583–13589, 2020.
- [24] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2011.
- [25] S. Thys, W. Van Ranst, and T. Goedemé. Fooling automated surveillance cameras: Adversarial patches to attack person detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 49–55. IEEE, 2019.
- [26] F. Vakhshiteh, A. Nickabadi, and R. Ramachandra. Adversarial attacks against face recognition: A comprehensive study. *IEEE Access*, 9:92735–92756, 2021.
- [27] Vanessa Volz, Jacob Schrum, Jialin Liu, Simon M Lucas, Adam Smith, and Sebastian Risi. Evolving mario levels in the latent space of a deep convolutional generative adversarial network. In *Proceedings of the genetic and evolutionary computation conference*, pages 221–228, 2018.
- [28] X. Wang, Z. Zhao, C. Zhang, N. Bai, and X. Hu. *SE-ResNet56: Robust Network Model for Deepfake Detection*, page 37–52, 2023.
- [29] C. Wu, W. Luo, N. Zhou, P. Xu, and T. Zhu. Genetic Algorithm with Multiple Fitness Functions for Generating Adversarial Examples. In *Congress on Evolutionary Computation*, pages 1792–1799, 2021.
- [30] C. Xiao, B. Li, J. Zhu, W. He, M. Liu, and D. Song. Generating adversarial examples with adversarial networks, 2018.