



UNIVERSITÀ
DEGLI STUDI
FIRENZE



UNIVERSITÀ
DEGLI STUDI
DI PERUGIA

INdAM | ISTITUTO NAZIONALE
DI ALTA MATEMATICA

University of Florence, University of Perugia, INdAM, consortium in CIAFM

Ph.D. in Mathematics, Computer Science, Statistics

Curriculum in Computer Science - CYCLE XXXV

Administrative office at University of Florence

One-for-Many:
A flexible adversarial attack on different
DNN-based systems

Academic Discipline (SSD) INF/01

Ph.D. Candidate

Dr. Alina Elena Baia

Supervisor

Prof. Valentina Poggioni

Coordinator

Prof. Matteo Focardi

Years 2019/2022

Contents

Introduction	4
1 Background	8
1.1 Image Adversarial Attacks	8
1.1.1 Image Quality Assessment	11
1.1.2 Defense methods	13
1.2 Evolutionary Algorithms	15
1.2.1 Genetic Algorithm	16
1.2.2 Differential Evolution	17
1.2.3 Evolution Strategies	18
1.3 Multi-Objective Optimization	19
2 Related Work	25
2.1 Image Classification	25
2.2 Object detection	27
2.3 Explainable AI	29
3 The <i>One-for-Many</i> attack	32
3.1 General approach	32
3.2 Image filters	33
3.3 The adversarial algorithm	34
3.3.1 Outer-optimization step	36
3.3.2 Inner-optimization step	39
3.3.3 Queries to the target model	40

4	Attacks on image classifiers	41
4.1	Per-instance Single Objective Attack	41
4.1.1	Problem formulation	42
4.1.2	Experimental setup	43
4.1.3	Evaluation and Experimental Results	44
4.2	Per-instance Multi-objective Attack:	
	Emotion Recognition	52
4.2.1	Problem formulation	53
4.2.2	Experimental setup	53
4.2.3	Evaluation and Experimental Results	55
4.3	Universal Multi-objective Attack	61
4.3.1	Problem formulation	61
4.3.2	Experimental Setup	62
4.3.3	Evaluation and Experimental Results	64
5	Attacks on object detectors	69
5.1	Introduction	69
5.2	Problem Formulation	70
5.3	Evaluation and Experimental Results	71
6	Attacks on multimodal explanations	80
6.1	Introduction	80
6.2	Problem formulation	81
6.3	Experimental setup	82
6.4	Evaluation and Experimental results	85
	Conclusions	95
A	Image filters	100
B	Ablation studies	103
B.1	Per-instance Single Objective attack	103

B.2	Per-instance Multi-objective attack: Emotion Recognition	104
C	Case study: Many-objective attack	107
C.1	From multi to many-objective problem	107

Introduction

Motivation

Deep Neural Networks (DNNs) have become the standard-de-facto technology in most computer vision applications due to the exceptional performance and versatile applicability they demonstrated in the last years. However, studies have shown that DNNs are remarkably vulnerable to adversarial examples, input data intentionally modified with perturbations specifically crafted to mislead a model into making wrong predictions.

Most of the proposed works on adversarial examples focus on finding small image perturbations bounded by L_p -norm distance measure, known as restricted attacks, forcing the adversarial example to be as similar as possible to the original data. Due to the urgency of taking counter-measures, several defense techniques have been introduced to overcome such vulnerabilities. Nowadays, most of the restricted perturbations can be defended with adversarial training or with input denoising and restoration.

To overcome the limitations of restricted attacks, unrestricted methods that allow unbounded large perturbations (i.e. geometric image transformation or color manipulation) have been recently proposed.

To learn image manipulation filters, the majority of such methods require full access to the target model, which is not always feasible, especially in real-world applications.

For this reason, algorithms that work in a black-box fashion have been recently introduced. Nevertheless, crafting unrestricted perturbations in a black-box setup where the adversary has no knowledge about the target model often results in suspicious unnatural images that do not deceive the human eye. Moreover, some methods require additional resources (i.e. image segmentation models) to extract prior information about the image and modify its

colors in accordance with human perception. In this case, the naturalness of an image and its non-suspiciousness highly depends on the performance of the segmentation network.

Although some works on unrestricted adversarial attacks have been proposed, the area of image filtering attacks and colorization-based methods is still under-explored. This further motivated us to conduct this study and to introduce novel approaches to address the limitations of existing methods and to expand the landscape of this research topic.

Contributions

In this work, we propose the *One-for-Many* attack, a black-box method to generate unrestricted adversarial perturbations by optimizing multiple Instagram-inspired image filters that manipulate specific image characteristics such as saturation, contrast, and brightness, perform edge-enhancement, or apply light gradient. By using well-known image manipulation filters available in several image processing libraries, modern cameras, and widely used in social media (e.g. Instagram, Facebook) we aim to reduce noticeability and to produce natural-looking adversarial examples without relying on additional resources. Moreover, the combination of filters is useful to generate more reliable and transferable perturbations and create images with a wide range of visual effects, including soft warm looks and vibrant colors.

The proposed method generates the adversarial perturbations with a two-step nested-evolutionary algorithm: given a set of parameterized image filters, the outer optimization step determines the sequence of filters to apply to an image, while the inner step optimizes the parameters of each filter selected in the previous step.

The algorithm is flexible and can be easily customized for many computer vision tasks. It also allows different attack strategies and combinations of multiple objectives, such as image-specific or universal attacks, and single or multiple-objective optimization.

We validate the proposed adversarial attack on state-of-the-art image classifiers, object detectors, and a newly proposed multimodal explanation model for activity recognition. The experimental results show that the method generates high-quality natural-looking adversarial images that can effectively fool the above-mentioned systems.

In the case of image classification, our method generates more transferable, more robust, and more deceitful adversarial perturbations than a similar state-of-the-art method. The proposed attack also greatly decreases the task performance of object detection models while also maintaining good transferability properties. These results indicate that more effort is necessary in order to increase the robustness of deep neural networks to common image editing techniques. On the other hand, by leveraging this vulnerability, our method could be employed for the development of privacy protection tools that apply customized image filters to defend the user’s privacy from unauthorized automatic information extraction on social media platforms.

Attacks to multimodal explainable systems are one of the newest emerging trends. To the best of our knowledge, no other work has explicitly studied the robustness of such systems in a black-box setting. In particular, we focus on a novel explanation model that takes an image as input, predicts an activity label, and generates a textual and visual explanation. The attack is effortlessly adapted to consider objective functions for different types of data (i.e. image and text data) and successfully breaks the correlation between activity prediction and its explanations under two scenarios: keeping the activity the same and changing the textual explanation, and vice-versa. The results obtained are very exciting and open up a line of research where our method could be used to develop a system-independent explanation evaluation metric to enable comparative analysis of different vision-language explanation systems, which the literature lacks at the moment.

We hope that our work will inspire further research and studies on the susceptibility of deep neural models to image filtering attacks and that our findings will help deepen the understanding of the implications posed by such attacks and encourage the development of robust defenses.

Organization of the thesis

This work is organized as follows: in the first chapter we introduce basic concepts about adversarial attacks and evolutionary algorithms; in the second chapter we provide a review of the related work; in the third chapter, we present the general structure of the proposed

algorithm and its main operations; we validate our attack on image classifiers under different configurations in Chapter 4 and on object detection models in Chapter 5; the case study on multimodal explanations is presented in Chapter 6. We conclude with some final considerations and remarks about future work.

Chapter 1

Background

In this chapter, we briefly introduce the basic concepts and algorithms to facilitate the comprehension of the following chapters. In Section 1.1 we present the notion of adversarial attacks, providing also a description of the image quality assessment metrics and defense methods used throughout the course of this work. Section 1.2 focuses on the evolutionary algorithms. We conclude with an introduction of multi-objective problems in Section 1.3.

1.1 Image Adversarial Attacks

Deep neural networks (DNNs) have achieved state-of-the-art performance in various computer vision tasks, including image classification [1–5], object detection [6–10], and image segmentation [11–15]. Moreover, with the increase of open source models and improvements on their usability, DNNs have become extensively used in real-world day-to-day applications and are nowadays integrated in safety and security-critical systems and environments (i.e. self driving cars, robots and drones, surveillance, smart home IoT solutions, threats detection and prevention, healthcare). Despite their effectiveness, many studies [16–36] have shown the vulnerability of DNNs to adversarial attacks that perturb natural inputs to create adversarial examples that lead well-trained DNNs to produce erroneous predictions. This poses significant security and privacy issues and raises great concern about the implications of adopting DNN-based solutions and services. As a result, there has been a lot of interest from the research community for creating innovative adversarial attacks in order to identify

the vulnerabilities of deep neural networks and to assess their limitations. Furthermore, the necessity to protect DNN models against such attacks has led to proposing defenses and countermeasures which help develop more robust models. While adversarial attacks might be considered a big threat, they provide fundamental value for a great good.

Formally, given an input image x and its corresponding label y , let M be a neural network model that maps x to y , that is $M(x) = y$. An adversarial attack alters the input image x with a perturbation δ to generate an adversarial example $x^* = x + \delta$ that induces the neural network to predict a different label: $M(x) \neq M(x^*)$.

In general, adversarial techniques, at the highest level, can be grouped by the level of knowledge and access to the target model, by the type of perturbations or by their adversarial goal.

Based on the type of adversarial perturbations, the attacks can be classified as *restricted* or *unrestricted*. In the restricted case, the amount of modification applied is bounded by a L_p -norm distance measure [16–25, 32, 36, 37], forcing the adversarial image x^* to be as close as possible to the original one. However, restricted perturbations are often not semantically meaningful and can create visible artifacts that can be detected by defenses [38–41]. On the other hand, unrestricted attacks do not limit the amount of change and use large perturbations without L_p -norm constraints which sometimes can result in overly distorted images [28]. To tackle this issue, unrestricted methods that manipulate basic image attributes (such as color, texture, contrast, saturation, and brightness) have been proposed to create photo-realistic natural-looking images [26, 29, 31, 42]. Such unrestricted adversarial examples have been found to be more transferable to unseen models and more robust to defense mechanisms than restricted ones [31].

Adversarial attacks can also be classified in *per-instance attacks* or *universal attacks*. In the first category, we can find all those systems that generate a different perturbation for each image; in that case, a separate optimization process has to run for each image in order to find an image-specific adversarial perturbation [16, 17, 19, 23, 25, 27, 31]. In the second category we can find all those systems able to find a unique universal perturbation that can fool the deep learning model when applied on 'any' image; these systems are called universal because they are essentially image-agnostic [33, 34, 43–47].

Considering the adversarial intent, attacks can be further distinguished in *targeted*, when the prediction of the neural model is misguided towards a specified target label, or *untargeted* when the goal is simply to generate an output different from the original one.

An attack can be classified into either *white-box* or *black-box* based on the amount of information available to the adversary about the target model. In a white-box scenario, the attacker has full access and knowledge about the target model (i.e. specific architecture and its parameters, training policy, and training data) and utilizes the available information to generate the adversarial example. On the contrary, in a black-box setup [21, 30, 31, 34, 42, 48–56] the adversary had no knowledge about the model and the only way to gain information is to query the target model and observe the output for given inputs. In this case, the attack should be query-efficient because there could be restrictions on the number of queries as a result of limits on different resources, such as a time limit or a monetary limit if the attacker incurs a cost for each query [30, 34, 56]. This makes black-box attacks more challenging but their applicability more practical to real-world applications.

Regardless of their nature, adversarial attacks should be effective, robust, transferable, unnoticeable, and undetectable [57]. The degree to which an adversarial attack is successful in deceiving a machine learning model determines its effectiveness. An attack is considered robust if it remains effective in the presence of defenses intended to mitigate or remove the malicious effect of the adversarial perturbations before passing the image to a neural model. The transferability measures the ability of a perturbation designed for one model to successfully fool another model different from the one that was used to craft it. In an unnoticeable attack, the adversarial perturbation is not recognized by a human observer and the content of the image is perceived in the same way as in the clean image. A very common approach to measuring the noticeability and quality of an adversarial image is to use automatic image quality assessment (IQA) metrics to quantitatively estimate the level of image degradation after the adversarial manipulation. Finally, an adversarial attack should be undetectable. We can define undetectability as the extent to which an adversarial attack can bypass a defense mechanism that was designed to identify if an image was tampered with. Given that deep learning systems may be equipped with protection methods, having low detectability makes the attack more likely to succeed.

We briefly present the IQA metrics employed in this work in Section 1.1.1 and the defense frameworks in Section 1.1.2.

1.1.1 Image Quality Assessment

Adversarial perturbations can create unnatural-looking images which cannot bypass a human judgment [28, 43, 53, 58, 59]. Therefore, Image Quality Assessment (IQA) techniques should be used to quantify the visual quality of an image by analyzing different characteristics like aesthetics, naturalness, or distortions [60–62]. Over the years, many different methods have been proposed. There are essentially two types of IQA methods, *subjective* and *objective*. Subjective assessment requires a human evaluation and intervention and is considered the most accurate and reliable. However, it is time-consuming, expensive, and impractical for real-time assessment applications.

Objective methods are designed to measure the visual quality of an image automatically fitting the human assessment. Using mathematical models or deep learning approaches, they prove highly efficient and ideal for image-based system optimization.

Based on the availability of the reference image, objective methods can be further divided into two main categories: *full-reference* (FR), and *no-reference* (NR). FR strategies require computing the quality score by comparing the modified image with the complete reference image. NR strategies, also known as blind assessments, are designed to accurately predict the image quality without using a reference image or any additional information, thus being suitable for applications where the reference image is not available.

Over the years many image quality assessments metrics have been proposed. However, there is no perfect metric to automatically evaluate the quality of images that fits every scenario. Therefore, we chose some of the most popular and used metrics in the community that have been found to perform well against different type of distortions and image manipulation techniques such as contrast alteration, tone modification, color changes, noise and blur distortions. Specifically, we use SSIM index [63] as FR metric, and we use NIMA [64], NIQE [65], and MANIQA [66] as NR metrics.

Structural Similarity Index Measure (SSIM), introduced by Wang et al. [63], is an

FR context-aware metric that quantifies the image degradation as perceived changes in the structural information. SSIM is inspired by the human visual system (HVS) and is capable of extracting and identifying structural information from natural scenes (i.e., images), deeply structured with significant dependencies between spatially closed pixels. Structural information represents the structure of objects in an image which are independent of contrast and luminance. Thus, SSIM is defined as a comparison function of contrast, luminance and structure computed over the image. By design, the metric satisfies the symmetry, boundedness and unique maximum property that assures an upper value of 1 if and only if the two images compared are identical.

Neural Image Assessment (NIMA) [64] is an NR image metric that uses a trained Convolutional Neural Network to predict both technical and aesthetic qualities of images, giving importance to factors like contrast, tone, composition, framing, and color palette. The model is trained on the AVA [67] dataset containing about 255k images rated based on aesthetic qualities by amateur photographers. Each AVA photo is scored by an average of 200 people and the image ratings range from 1 to 10, with 10 being the highest aesthetic score associated to an image. The authors show that the aesthetic evaluation of NIMA closely matches the scores assigned by human raters and that it has a high correlation to human perception.

Natural Image Quality Evaluator (NIQE) [65] is an NR image quality assessment metric that uses only statistical information derived from natural images to predict how natural a given image is. NIQE does not use subjective quality scores and it does not require a training phase on human-rated large datasets or any exposure to distorted images. Instead, it uses a natural scene statistic model (NSS) to create a quality-aware collection of statistical features derived from a set of natural, unaltered images. Then, the quality of a given test image is computed as the distance between the NSS features extracted from the input image and the quality-aware features previously extract from the dataset of natural images. NIQE has been shown to be highly correlated with human perception and to achieve comparable results with NR IQA trained on human judgments of known distorted images.

Multi-dimension Attention Network for no-reference Image Quality Assessment (MANIQA) [66] is the new state-of-the-art NR image quality assessment metric and winner of the NTIRE2022 NR-IQA challenge [68]. It uses a transformer-based architecture to predict the perceptual quality of images in accordance with human subjective perception. The MANIQA model is trained on the PIPAL [69] dataset which contains images processed by image restoration and enhancement methods (particularly generative adversarial network-based methods) besides the traditional distorting methods (i.e blur, noise, compression). The dataset contains 29k images that cover 40 different distortion types and 116 distortion levels, involving 1.13 million human judgments. Experimental results and other studies [70] demonstrate that MANIQA outperforms state-of-the-art methods by a large margin.

1.1.2 Defense methods

The robustness of DNN against adversarial examples has gained significant attention in the last few years, and several approaches and systems able to defend from adversarial attacks have been proposed and developed. Some of them follow the *adversarial training* approach increasing the network robustness by means of adversarial examples in the training process [16, 18, 25, 71], while others propose *defense by detection* (i.e. feature squeezing [38], perturbation rectifying [72]), *defense by sanitization* that remove the effect of adversarial perturbations [39, 73–76] or ad-hoc trainable techniques like distillation to reduce the model sensitivity to small perturbations [41]. In our work, we explore Feature Squeezing [38] as detection method and Instagram Filter Removal-Net framework as sanitization method to mitigate the adversarial effect of image filters. Feature Squeezing is a highly performant metric able to generalize well across many state-of-the-art attacks. On the other hand, Instagram Filter Removal-Net specifically focuses on removing the effects of Instagram image filters which makes the method the perfect antagonist to our attack. Choosing different methods allows us to better evaluate the robustness of our adversarial perturbation.

Feature Squeezing is one of the most popular detection frameworks that achieves high detection rates against state-of-the-art attacks. This method is based on the observation that often the space of feature vectors of images is unnecessarily large which gives plenty

of manipulation possibilities for generating adversarial examples. The authors proposed to squeeze out unnecessary input features in order to reduce the search space accessible to an adversary by means of two feature squeezing methods: color bit depth reduction of each pixel and spatial smoothing (local and non-local smoothing). Using such input transformations, an image is tagged as adversarial if the L_1 difference between the prediction vectors of the original image and its squeezed version exceeds a certain threshold. The authors also show that adversarial examples from eleven state-of-the-art attacks can be successfully detected by combining multiple squeezing defenses into a joint detection framework. Given that the selection of an optimal threshold value is not a trivial task and requires a training phase, for the experiments in this work we refer to the thresholds reported by the authors in [38].

Instagram Filter Removal-Net (IFRNet) solves the problem of removing Instagram filters from the images as a reverse style transfer problem, where any visual effect injected by a particular filter is removed from an image by directly reverting them back to its original style. IFRNet consists of an encoder-decoder architecture: the encoder is used as a style extractor module and uses an adaptive feature normalization strategy in all layers of the encoder to eliminate the external style information; then the normalized features are fed into the decoder that generates the unfiltered version of the input image with the help of adversarial training, inspired by generative adversarial networks. The IFRNet is trained on the IFFI dataset [73] that contains 9600 high-resolution and aesthetically pleasing images along with their filtered versions by 16 different Instagram filters (i.e. Clarendon, Hudson, Perpetua, Gingham, Valencia, and more). Experiments on the Instagram Filter Removal task verify that IFRNet eliminates external visual effects to a great extent.

1.2 Evolutionary Algorithms

Evolutionary Algorithms (EAs) are population-based meta-heuristic optimization methods inspired by biological evolution. EAs maintain a group of solutions, called a population, to optimize or learn the problem. The population is a basic principle of the evolutionary process. Every solution in a population is called an individual and every individual is evaluated by a fitness function, a quality metric that measures the performance of a solution. EAs prefer fitter individuals, which is the basis for algorithm optimization and convergence. New candidate solutions are generated using a number of variation operations (i.e. recombination and mutation) and the best individuals are passed on to the next generation. This process is iterated over multiple generations until a stopping criterion is met. The goal is to evolve a population over time and identify better solutions. The general scheme of an evolutionary algorithm is given in Algorithm 1.

In this work, three EAs have been considered: Genetic Algorithm [77], Differential Evolution [78], and Evolution Strategies [79]. We briefly present each of them in the following sections.

Algorithm 1 General scheme of an Evolutionary Algorithm

- 1: Initialize a population with random candidate solutions
 - 2: Evaluate initial population
 - 3: **repeat**
 - 4: Select parents
 - 5: Recombine pairs of parents
 - 6: Mutate the resulting offspring
 - 7: Evaluate new candidates
 - 8: Select individuals for the next generation
 - 9: Update the current population with the selected individuals
 - 10: **until** termination condition is satisfied
-

1.2.1 Genetic Algorithm

Genetic Algorithms (GA) are a family of population-based heuristic search approaches commonly applied in the literature to a variety of optimization problems. The population is composed of a set of candidate solutions for the optimum of the problem; the representation of a solution determines the applicability of genetic operators. In this work, we use integer vectors to represent the sequence of filters, while the parameters of the filters in each sequence are assembled in real-valued vectors. GA evolves a population of solutions towards an optimal solution by means of crossover, mutation, and selection operators.

Crossover: The crossover operator determines how the genetic material of the population elements is combined to obtain a new offspring. The most common operator is n-point crossover: two population elements are split at n-points and an offspring is built by alternately picking sequences from the two parents. Additional strategies for continuous solutions include aggregating component-wise, e.g. computing the average values, or picking each component from one of the parent solutions, randomly or with a fixed percentage of the solution length. We adopt 1-point crossover in our implementation.

Mutation: The mutation operator randomly perturbs offspring to further explore the solution space. For example, a common mutation strategy for bit-strings is random bit flipping, mutating each component with according to a fixed probability. For continuous solutions, random noise is applied to each component, e.g. Gaussian noise.

Selection: After computing new offsprings, each individual is evaluated based on a fitness value. Then, a selection operator is applied to choose which candidates become part of the new parental population. Elitist selection operators choose the best solutions according to their fitness value; randomness-based operators, e.g. roulette wheel or tournaments, are instead not guaranteed to choose the best performing-based solutions, to overcome local optima and favor the exploration of solution space. In our algorithm, we adopt elitist selection.

Update: During the last step, individuals from the current population are replaced by the new selected individuals and the new generation is formed. The above described operations are repeated until a stopping condition is satisfied (i.e the algorithm reached the allowed number of generations or the fitness function has attain a predefined value).

When using genetic algorithms it is important to be aware that finding solutions to certain problems might require high number of generations and a large population size which impact the time complexity. In general, the time complexity of a genetic algorithm depends on several factors, such as the number of generations, population size, the type of genetic operators, the selection strategies, and the complexity of fitness evaluation. The fitness evaluation depends on the nature of problem being solve and often it represents the most computationally expensive step: it can vary from simple calculations to more complex computations, like neural network inference.

Thus, assuming a population size of N , a number of generations G , the time complexity of a genetic algorithm can be roughly estimated by $O(G \times N \times F)$ where F represents the complexity of the fitness function evaluation. Overall, it is challenging to precisely determine the complexity of genetic algorithms due to their stochastic and problem-specific nature and in general the complexity is computed in terms of fitness function evaluations. Therefore, an alternative, better way to assess the complexity of the algorithm is to measure the running time.

1.2.2 Differential Evolution

Differential Evolution (DE) [78] is a population-based optimization algorithm designed for optimization over continuous spaces and does not require the optimization function to be differentiable or linear. Similar to GA, DE iteratively evolves a population of candidate solutions evaluated by means of a fitness function. Unlike GA, it creates a new individual by first performing the mutation operation and then crossover.

Mutation: The mutation operation generates a mutant vector d , called *donor vector*, through a differential operation performed on vectors of the current population. In literature, several mutation strategies have been proposed that consider different numbers of vectors for differential mutation and different selection schemes. In this work we use *rand/1* mutation in combination with binomial crossover since it has been found to perform the best in a variety of optimization problems [80, 81]. Specifically, for each *target vector* x_i of the

current population, a donor vector v_i is computed as follows:

$$v_i = x_a + F * (x_b - x_c) \quad (1.1)$$

where x_a, x_b , and x_c are three randomly selected members of the current population, with $a \neq b \neq c \neq i$ and $F \in [0, 2]$ is a user-specified mutation factor.

Crossover: The crossover operator creates a new vector u_i , denoted as *trial vector*, by combining the genetic information of the target and donor vector using a binomial crossover defined as:

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } rand_{ij} \leq CR \text{ or } j = j_{rand} \\ x_{i,j} & \text{otherwise} \end{cases} \quad (1.2)$$

For each decision variable j of the trial vector, a random number $rand_{ij}$ in the range $[0,1]$ is generated. The decision variable at index j of u_i inherits from the donor vector v_i if the $rand_{ij}$ is smaller than a user-specified crossover probability (CR), otherwise, it inherits the information from the target vector x_i . The parameter j_{rand} , an integer random number in $\{1, \dots, d\}$ with d being the dimension of the vectors, guarantees that at least one decision variable is inherited from the donor vector v_i .

Selection: The selection operator compares each trial vector with a target vector and the individual with the better fitness is selected for the next generation. We use one-to-one selection where a trial vector u_i is compared directly with its corresponding target vector x_i .

1.2.3 Evolution Strategies

Evolution Strategies (ES) is a family of population-based meta-heuristics inspired by natural evolution and designed specifically for continuous function optimization. The original version proposed by Rechenberg et al. [79] included the $(1 + 1)$ and $(1, 1)$ strategies. In the former, a population member is replaced only with offspring having better fitness function value; in the latter, an offspring always replaces its parent to favor exploration and avoid getting stuck in local optima.

Differently from the GA and DE, ES uses only mutation and does not use any form of crossover operation. New candidate solutions are generated by perturbing the genetic information of the parents with a random noise sample from a Gaussian distribution. The

peculiarity of ES lies in the concept of self-adaptivity, introduced in more refined versions of the algorithm such as $(\mu + \lambda)$ and (μ, λ) that follow the original strategy, where μ is the number of members in the population and λ the number of offsprings generated in one cycle. In this work, we use a variant of ES introduced in [82].

Given an initial individual, a batch of λ samples are generated using mutation and the fitness value of each sample is evaluated. The fitness values are then used to calculate a gradient estimate towards a better solution and update the original individual (Algorithm 2). We refer to this as $(1, \lambda)$ -ES.

Algorithm 2 $(1, \lambda)$ -Evolution Strategies

Require: learning rate α , decay rate β , noise standard deviation σ , initial individual p of

parameters θ_0

- 1: **for** $t = 0, 1, 2, \dots$ **do**
 - 2: sample $\epsilon_1, \dots, \epsilon_n \sim \mathcal{N}(0, \sigma)$
 - 3: compute fitness $F_i = F(\theta + \sigma\epsilon_i)$ for $i = 1, \dots, n$
 - 4: set $\theta_{t+1} \leftarrow \theta_t + \alpha \frac{1}{n\sigma} \sum_{i=1}^n F_i \epsilon_i$
 - 5: $\sigma = \sigma * \beta$
 - 6: **end for**
-

1.3 Multi-Objective Optimization

In multi-objective optimization, the aim is to solve problems of the type¹:

$$\text{minimize } \vec{f}(\vec{x}) := [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})] \quad (1.3)$$

subject to:

$$g_i(\vec{x}) \leq 0 \quad i = 1, 2, \dots, m \quad (1.4)$$

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \dots, p \quad (1.5)$$

where $\vec{x} = [x_1, x_2, \dots, x_n]^T$ is the vector of decision variables, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, k$ are the objective functions and $g_i, h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$, $j = 1, \dots, p$ are the constraint

¹Without loss of generality, we will assume only minimization problems.

functions of the problem.

Definition 1. Given two vectors $\vec{x}, \vec{y} \in \mathbb{R}^k$, we say that $\vec{x} \leq \vec{y}$ if $x_i \leq y_i$ for $i = 1, \dots, k$, and that \vec{x} **dominates** \vec{y} (denoted by $\vec{x} \prec \vec{y}$) if $\vec{x} \leq \vec{y}$ and $\vec{x} \neq \vec{y}$.

Definition 2. We say that a vector of decision variables $\vec{x} \in \mathcal{X} \subset \mathbb{R}^n$ is **non-dominated** with respect to \mathcal{X} , if there does not exist another $\vec{x}' \in \mathcal{X}$ such that $\vec{f}(\vec{x}') \prec \vec{f}(\vec{x})$.

Definition 3. We say that a vector of decision variables $\vec{x}^* \in \mathcal{F} \subset \mathbb{R}^n$ (\mathcal{F} is the feasible region) is **Pareto-optimal** if it is non-dominated with respect to \mathcal{F} .

Definition 4. The **Pareto Optimal Set** \mathcal{P}^* is defined by:

$$\mathcal{P}^* = \{\vec{x} \in \mathcal{F} | \vec{x} \text{ is Pareto-optimal}\}$$

Definition 5. The **Pareto Front** \mathcal{PF}^* is defined by:

$$\mathcal{PF}^* = \{\vec{f}(\vec{x}) \in \mathbb{R}^k | \vec{x} \in \mathcal{P}^*\}$$

When solving multi-objective optimization problems (MOPs), the aim is to obtain the Pareto optimal set from the set \mathcal{F} of all the decision variable vectors that satisfy (2) and (3). Thus, in a MOP, the goal of a Multi-Objective Evolutionary Algorithm (MOEA) is to produce a good approximation of its Pareto front.

In this work we use the *crowding-comparison operator* (Algorithm 4) and *non-dominated sorting* (Algorithm 3) of the elitist Non-dominated Sorting Genetic Algorithm (*NSGA-II*) [83,84] as selection strategy for multi-objective problems. NSGA-II is one of the most widely used MOEAs for problems having two or three objectives.

Given S a set of parents and offsprings, and \mathcal{F}_{MO} a problem specific multi-objective fitness, the *non-dominated sorting* procedure divides the set of points $P = \{\mathcal{F}_{MO}(s) | \forall s \in S\}$ into *non-dominated fronts* \mathcal{PF} according to *non dominance* relation (Algorithm 3). Then, for each individual in the same *non-dominated* front computes the *crowding distance* (Algorithm

4). Finally, all individuals of all fronts are combined in a single set \bar{P} and sorted using the following partial order:

$$i \prec_n j \equiv i_{rank} < j_{rank} \vee (i_{rank} = j_{rank} \wedge i_{distance} < j_{distance}) \quad (1.6)$$

where i_{rank} is the value computed in the *non-dominated sorting* step, and $i_{distance}$ is the *crowding distance*. After obtaining the sorted set \bar{P} (Algorithm 5), in our multi-objective applications we select the best k solutions to pass to the next generation in the optimization process.

The time complexity of the *non-dominated sorting*, *crowding distance* and *sorting on \prec_n* is $O(M(2N)^2)$, $O(M(2N) \log(2N))$ and $O(2N \log(2N))$, respectively, where M is the number of objectives and N is the population size. Thus, the overall complexity is $O(MN^2)$ where the dominant factor is the computational cost of sorting the population based on non-domination. Nevertheless, in this work, the cost of this procedure is fairly low since we consider only two objectives (a preliminary study with three objective is presented in the Appendix C) and the population size is kept small.

For the sake of clarity, an overview of the NSGA-II procedure is shown in Figure 1.1. Suppose S_t is a population of size $2N$ formed by the current population P_t and the current offspring Q_t . After the entire population S_t is evaluated based on the objective function, the population S_t is sorted according to the non-dominance relationship and divided into a series of non-dominated fronts $F_1, F_2, F_3, \dots, F_l$ by assigning a rank to each individual. The lower the rank, the better the individual is. This means that solutions from the non-dominated set F_1 represent the best solutions (the best trade-off between objectives) in the combined population and must be prioritized. The subsequent fronts represent progressively worse solutions. If the size of the first front F_1 is smaller than the population size N then all individuals of F_1 will be selected for the new population P_{t+1} . To complete the population P_{t+1} , the remaining individuals are chosen from subsequent non-dominated fronts in order of their ranking: individuals from F_2 are selected next, followed up by individuals from F_3 and so on until N individuals are chosen and the population P_{t+1} is created.

Usually, the number of individuals in all fronts is bigger than the population size and not all individuals can be inserted in the new population. Therefore, to select exactly N

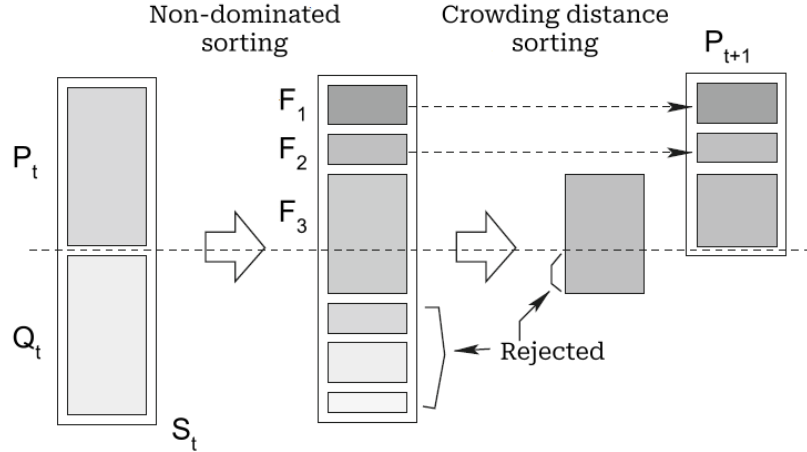


Figure 1.1: NSGA-II procedure. Source [83]

individuals, the solutions of the last front F_l are sorted using the crowding distance operator. Individuals with the highest crowding distance are chosen to fill in the available slots in P_{t+1} . The crowding distance measures the density of individuals in the objective space and helps to maintain diversity. It is calculated by estimating the perimeter of the cuboid formed by using the nearest neighbors as the vectors. Figure 1.2 illustrates an example of crowding distance computation. The crowding distance of solution i is the average side length of the cuboids. Moreover, it is important to note that the borders of the front have an infinite crowding distance which means they are always preferred in the selection phase. Crowding distance can be seen as an estimation of the density of solutions around a particular solution in the population. Solutions that are situated in less dense regions are preferred in order to improve diversity. Finally, after selecting individuals based on their crowding distance the newly created population P_{t+1} is passed on to the next generation.

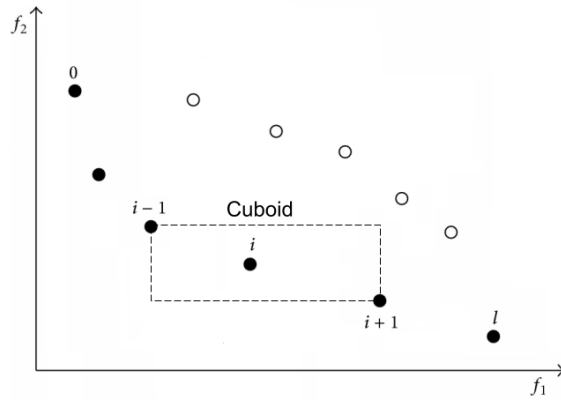


Figure 1.2: Crowding distance calculation. Points marked in filled circles are solutions that belong to the same non-dominated front. Source [83]

Algorithm 3 Non-Dominated Sorting

Require: set of points P

- 1: **for** $p \in P$ **do**
 - 2: $S_p \leftarrow \emptyset, n_p \leftarrow 0$
 - 3: **for** $q \in P$ **do**
 - 4: **if** $p \prec q$ **then** $S_p \leftarrow S_p \cup \{q\}$
 - 5: **else if** $q \prec p$ **then** $n_p \leftarrow n_p + 1$
 - 6: **end for**
 - 7: **if** $n_p = 0$ **then** $p_{rank} \leftarrow 1, \mathcal{F}_1 \leftarrow \mathcal{F}_1 \cup \{p\}$
 - 8: **end for**
 - 9: $i \leftarrow 1$
 - 10: **while** $\mathcal{F}_i \neq \emptyset$ **do**
 - 11: **for** $p \in \mathcal{F}_i$ **do**
 - 12: **for** $q \in S_p$ **do**
 - 13: $n_p \leftarrow n_p - 1$
 - 14: **if** $n_p = 0$ **then** $q_{rank} \leftarrow q_{rank} + 1, \mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$
 - 15: **end for**
 - 16: **end for**
 - 17: $i \leftarrow i + 1$
 - 18: $\mathcal{F}_i \leftarrow \mathcal{Q}$
 - 19: **end while**
 - 20: **return** $(\mathcal{F}_1, \mathcal{F}_2, \dots)$
-

Algorithm 4 Crowding Distance

Require: set of non-dominated fronts $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \dots)$

```
1: for  $\mathcal{I} \in \mathcal{F}$  do
2:    $l \leftarrow |\mathcal{I}|$ 
3:   for  $i \leftarrow 1 \dots l$  do
4:      $\mathcal{I}[i]_{distance} \leftarrow 0$ 
5:   end for
6:   for  $m \leftarrow 1 \dots N_m$  do
7:     sort  $\mathcal{I}$  by  $m$ -th objective function
8:      $\mathcal{I}[1]_{distance} \leftarrow \infty, \mathcal{I}[l]_{distance} \leftarrow \infty$ 
9:     for  $i \leftarrow 2 \dots (l - 1)$  do
10:       $\mathcal{I}[i]_{distance} \leftarrow \mathcal{I}[i]_{distance} + \frac{\mathcal{I}[i+1].m - \mathcal{I}[i-1].m}{f_m^{max} - f_m^{min}}$ 
11:    end for
12:  end for
13: end for
```

Algorithm 5 Crowding distance sorting

Require: set of nondominated fronts $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \dots)$

```
1:  $\bar{P} \leftarrow \emptyset$ 
2: for  $i = 1 \dots l$  do  $\bar{P} \leftarrow \bar{P} \cup \mathcal{F}_i$ 
3: sort  $\bar{P}$  by  $\prec_n$ 
4: return  $\bar{P}$ 
```

Chapter 2

Related Work

In this chapter we provide a synthesis of the existing literature that is most relevant to our work. We present the main characteristics of the state-of-the-art methods and highlight the limitations that our study aims to address. In Section 2.1 we focus on adversarial attacks against image classifiers, in Section 2.2 we introduce the related work on object detectors and we conclude with an overview of attacks against explainable AI systems in Section 2.3.

2.1 Image Classification

Many methodologies have been proposed for generating adversarial examples on image classifiers in both white-box and black-box settings. Most of the proposed works have been focusing on finding L_p -norm restricted attacks to generate imperceptible perturbations [16–21, 23–25, 32, 36, 37, 43, 50, 51, 53, 54, 85, 86]. However, restricted attacks have limited robustness to adversarial defenses and also limited transferability capabilities [29, 31, 38–40, 76, 87–89].

As a result, studies on unrestricted adversarial attacks have been emerging, such as geometric transformation attacks [90, 91], semantic attacks [92–94] and colorization attacks [26, 28, 29, 31, 95, 96]. Geometric attacks are easily noticeable due to the image distortion introduced by transformations such as rotation or translation. Semantic attacks change the semantics of an image by adding new content to the image (i.e. sunglasses to facial images to fool face recognition models or changing the weather conditions illustrated in an image to fool autonomous navigation systems). Considering the uniformity of the perturbation appli-

cation, colorization attacks remain a valid approach to craft non-suspicious natural-looking images. The common approach to generating adversarial examples is by changing the basic attributes of an image such as contrast, saturation, and brightness using image filters or by changing the colors.

Most attacks perform in a white-box setup: ACE [97] optimizes an adversarial color transformation filter similar to Instagram filters using gradient information, ALA [98] creates filters to manipulate the light in an image, FilterFool [96] trains a neural network to imitate traditional image processing filters (i.e. gamma correction), while other methods train neural networks to change the colors or the texture [26].

On the other hand, in a black-box setup, SemanticAdv [28] generates adversarial images by randomly changing the hue and saturation values of an image in the HSV color space while maintaining the shape of the objects. However, SemanticAdv does not consider the content of the image and often generates unnatural colors. ColorFool [31], a state-of-the-art method, proposes to improve the quality of adversarial images by perturbing the colors based on the semantics of an image. First, it identifies non-sensitive and sensitive regions using a semantic segmentation model. Then it changes the colors of each region in the Lab color space by adding adversarial perturbations in channels a and b while keeping the L channel unaltered. For the non-sensitive regions, the perturbations are chosen randomly from the entire range of possible values, whereas for the sensitive regions, the perturbation is selected randomly from a natural color range specifically defined for each region based on the region semantics and human color perception. In this case, the naturalness of adversarial images is strongly related to the accuracy of the segmentation model.

An interesting case study of image classification is represented by emotion recognition. Adversarial attacks on emotion recognition is a very recent application and just very few works are available in the literature [92,99,100]. The main difference with our work relies on the approach: white-box versus black-box. Since our algorithm works in a black-box scenario, it does not require any information about the model’s parameters or gradient values, as the other systems require. Hence, our approach can be applied against any system without having any knowledge about it. Moreover, they also differ in the way the images are modified. In particular [92] belongs in the category of physical attacks since it realizes attacks to facial

biometric systems by printing a pair of eyeglass frames. In [99,100] a saliency map extractor is used to extract the essential expression features of the clean facial expression example and a face detector is employed to find the position of the face in the image. This information is then used to enhance and cut the gradient of the input samples computed by the optimized momentum iterative method (OMIM) with respect to the misclassification loss.

Moreover, universal adversarial attacks have also been introduced. The authors [43] proposed to iteratively accumulate restricted image-specific perturbations until a certain percentage of input images are misclassified. Other approaches involve using Generative Adversarial Networks to model the distribution of universal perturbations [45, 46, 101–103]. Among these, [45] present a generalizable and data-free approach for crafting universal adversarial perturbations that can fool the target model without any knowledge about the data distribution, such as, the number of categories, type of data or the data samples themselves. Multi-objective evolutionary adversarial attacks have also been proposed [54, 55, 104]. They aim to simultaneously maximize the attack success rate and limit the perturbation applied with L_p -norm measures.

Differently from these, in one of our case studies we focus on unrestricted universal perturbations, and we include in the optimization process, alongside the maximization of the attack rate, the minimization of the detection rate of defence methods in order to produce attacks that intrinsically have the ability to bypass defenses.

2.2 Object detection

An object detector is a model designed to identify and locate objects within an image or a video. It is a fundamental component in many applications, such as autonomous driving, surveillance systems, robotics, and image analysis. Recently, some attack techniques have been introduced to craft adversarial images against object detection models.

This kind of attack was systematically studied for the first time in [105] with the algorithm DAG. The authors proposed an iterative white-box method that tries to assign adversarial labels to each region of interest in the image by adding noise to the original image. It runs gradient backpropagation to minimize a loss function computed as the sum, with respect

to all the targets in the image, of the differences between the score assigned to the original correct class and the one assigned to the adversarial incorrect class.

Co et al. proposed in [59] to create attacks to object detection systems by applying procedural noise functions to the original images, in particular Perlin noise and Gabor noise. They hypothesized that procedural noise, which exhibits patterns visually similar to the Universal Adversarial Perturbations (UAPs) proposed by [43], can also act as UAPs. They empirically demonstrated the vulnerability of some Deep Convolutional Networks to this procedural noise, and demonstrated that the same procedural noise is able to attack also Yolo-v3 [106] object detector on the MS COCO [107] dataset.

In [108] the authors proposed a black-box attack that finds a restricted adversarial perturbation using an evolutionary-based optimization. Specifically, they use Particle Swarm Optimization (PSO) and regard an image as a particle in a high-dimensional space. They generate the initial image by adding random noise to the original image and continuously optimize the image particle to find the adversarial image and also optimize its quality by moving it towards the original image. Li et al. [109] presented a black-box method to generate restricted adversarial perturbation that successfully attacks the Region Proposal Network, a component used in many object detectors. In [110] the authors propose a query-based black-box attack that searches for L_p -bounded rectangular perturbations in regions having a higher probability to contain objects. This results in effective attacks to object detection systems that use different DNNs as the backbone.

A different approach is represented by the *Dispersion Reduction* method proposed in [111]. The authors' idea is to transfer the concept of image "contrast" into the feature maps produced by a convolutional neural network. As lowering the contrast of an image can make the objects depicted unrecognizable, they proposed to reduce the contrast of an internal feature map to degrade object detection. They use a source model to craft the adversarial examples and then transfer it to object detectors. They formally defined the problem as a minimization problem of the dispersion (they use standard deviation) of the intermediate feature map of the modified image constrained by the L_∞ -norm with respect to the original one. Other approaches generate restricted perturbations using Generative Adversarial Networks (GANs) [24, 112].

We have to note that all the methods described above share the same general structure: optimized noise, restricted by L_p -norms, is added to the original images in order to obtain effective attacks. They differ in the noise applied and in the optimization algorithm used but the high-level idea is the same. Our method leverages the effective abilities of Instagram-inspired image filters to alter image attributes that allow the creation of non- L_p bounded adversarial images.

2.3 Explainable AI

Most deep neural models are black-box systems whose decision-making process is obscure. Explainable artificial intelligence (XAI) aims to make decisions of deep neural models transparent [113]. Effective XAI helps build trust in and accountability for AI decision-making processes. XAI systems also favour interactions for people and AI to jointly make decisions [114], assess vulnerabilities of a model [115], and identify biases [116, 117]. In this thesis, we focus on XAI for image classification tasks. Thus, we provide a review of the works related to this task.

Explainable artificial intelligence (XAI) approaches may generate visual, textual, or multimodal explanations. Visual explanations (V-XAI) highlight the most relevant pixel information used by the model [117–119]. Examples include saliency maps [118], heatmaps [117], super-pixels-based visualizations (e.g. LIME [119]), and feature contribution methods inspired by game theory (e.g SHAP [120]). However, V-XAI visualizations may be difficult to comprehend, especially for non-expert users, especially when no information is provided on how highlighted pixels influence the prediction. Textual explanations (T-XAI) describe the reasons for a decision in a more human-interpretable way (natural language sentences) [121–125]. Finally, multimodal explanations (M-XAI) jointly generate textual rationales and visual evidence in form of attention maps [126–128]. A recent M-XAI method [126] simultaneously predicts an answer (the prediction) and justifies, textually and visually, what led to that prediction (the explication).

Recent studies have shown that V-XAI models are susceptible to adversarial attacks that may, for example, preserve the prediction of the original image but change the explana-

tion [129–131]. Examples of attacks include restricted adversarial perturbations [129], structured manipulations that change explanation maps to match an arbitrary target map [130], adversarial classifiers that fool post-hoc explanations methods such as LIME and SHAP [131].

While several studies covered V-XAI methods, no approach has yet considered explicitly textual or multimodal explanations models. Related work on attacks to vision-language models for image captioning and visual question answering use restricted L_p norms bounded perturbations and operate under a white-box setup, with the attacker having full knowledge and access to the target model [103, 132–139] or in a gray-box setting with less information [103, 136, 140]. Attacks on image captioning may treat the structured output as a single label and design the attack as a targeted complete sentence [132]. This idea was extended to target keywords attacks that encourage the adversarial caption to include a predefined set of keywords [133] in any order [133, 135] or at specified positions in the caption [134]. Methods may mask out target keywords while preserving the caption quality for the visual content [138]. Untargeted attacks may use attention maps of the underlying target model to focus the adversarial noise in the regions attended by the model [139]. Specially designed generative models may also be used to generate adversarial perturbations [103, 136]. Alternatively, adversarial images may be generated by perturbing an image so that its features resemble the features of a target image and thus be indistinguishable from the model forcing it to output the same caption [140]. Multimodal networks are also vulnerable to adversarial perturbation on a single modality (i.e. the image input modality [141]). This idea was extended with a collaborative multimodal adversarial attack that performs the attack on both the image and text modalities [142] with the goal of changing the output of vision-language models (i.e. predicted label). These methods perform a white-box attack.

Unlike the methods mentioned above that generate white-box or gray-box restricted adversarial perturbations for CNN+RNN architectures for caption generation, we perform black-box, content-based attacks considering a transformer-based multimodal explanation system, that takes in input an image, predicts an activity, and generates a textual and visual explanation (see Fig. 6.1-6.2). Our attack uses only the final output (i.e. textual explanation or/and visual maps) of the model to find the adversarial perturbations that mislead the model which is more similar to a real-world scenario making the attack itself more challeng-

ing. Moreover, our case study is different from the other attacks to multimodal networks. The multimodal models used in [141] and [142] use different input modalities to solve tasks such as classification [141] or visual reasoning [142] where the goal is to predict whether the relationship between an image and a text is entailment, neutral, or contradiction. The purpose of the attacks is to change the final output of these models, which consists of a label. In our case study, the model under evaluation uses only the image input modality and returns a multimodal output composed of a prediction, a textual and a visual explanation. Our goal is to attack in a black-box setup one of the mechanisms only (i.e the prediction part) while keeping the other unchanged (i.e. the textual explanation part).

Chapter 3

The *One-for-Many* attack

In this chapter, we present the *One-for-Many* algorithm to generate adversarial examples. We provide a general description of the attack in Section 3.1. We introduce the image filters used for crafting the adversarial perturbation in Section 3.2. We present the structure and the main operations of the algorithm in Section 3.3.

3.1 General approach

We propose a highly flexible method to craft unrestricted adversarial examples in a pure black-box setting where the attacker has access only to the hard-label (the predicted label) by querying the target model and has no knowledge about the logits or the probabilities associated with the predicted labels.

We generate the adversarial examples by applying a composition of Instagram inspired parameterized image filters that manipulate the attributes of an image such as contrast, saturation, brightness, and sharpness, perform edge enhancement, gamma correction, or apply soft light gradients. This was inspired by the increasing popularity of social media platforms and photo editing apps which resulted in more and more people modifying their photos to achieve a desired look or aesthetic before sharing them online. And nowadays, the majority of images undergo some level of manipulation and post-processing filtering. Thus, we wanted to investigate if such common photo editing practices can be used to craft adversarial attacks. Our goal is also to assess the sensitivity of deep neural networks towards

image filtering since the literature lacks such studies.

Moreover, by using common filters in image processing libraries and widely used on social media we aim to reduce human awareness of image modifications. To find the successful adversarial filter configuration that misleads a target model, we designed a nested-evolutionary algorithm with outer and inner optimization steps. Given a predefined set of image filters, the outer optimization step focuses on finding the sequence of filters to use, while the inner step optimizes the parameters of each filter selected in the previous step.

Moreover, thanks to the use of a multi-objective evolutionary approach, we can adapt our method to attack different computer vision problems, ranging from image classification to textual explanations. The use of evolutionary optimization allows us to use any non-differentiable objective function, while multi-objective optimization allows us to merge different objectives, like for example the attack accuracy combined with image quality or attack detectability. This is our *One-for-Many* approach: one method for many problems, one algorithm for many different objectives.

3.2 Image filters

Photo editing has become a common practice in many business areas and particularly on social media sites where image enhancement and manipulation are extensively and excessively used. Thus, inspired by Instagram and Photoshop which offer tools to seamlessly modify images, we propose to combine multiple image filters in order to create custom adversarial image transformations. This approach provides plenty style options, ranging from subtle and warm looks to more dramatic and vivid color effects.

We implemented ten of the most popular Instagram filters: *Clarendon*, *Juno*, *Reyes*, *Gingham*, *Lark*, *Hudson*, *Slumber*, *Stinson*, *Rise*, and *Perpetua*. Each filter has distinct characteristics and effects given by the different levels of contrast, saturation, brightness, sharpness, edge enhancement, gamma correction, or soft light gradients.

For instance, *Clarendon* brightens and highlights a photo; *Gingham* gives a dusty-vintage feel to the image and it significantly lowers the highlights and the saturation; *Juno* adds saturation and warmth making the colors more intense; *Reyes* also adds a subtle old-time

look by reducing the saturation and by brightening up the photos; *Lark* increases the exposure making the photo brighter and reducing the vibrance; *Hudson* bumps up the blues giving a colder feel by applying a radial gradient with a dark blue exterior color and light interior color; *Slumber* desaturates colors, covering images with a soft haze by blending the original image with a plain light brown image; *Stinson* uses blending soft light to add more warmth to the image and increases the brightness and contrast; *Rise* gives a warm glow by mixing a radial gradient with a light sepia tone; *Perpetua* applies a vertical linear gradient that goes from yellow to blue over the image.

The application of each filter is regulated by two parameters, α and β . Each filter first scales the intensity of its specific basic attributes by a factor β and then applies the alpha blending (with parameter α) to obtain the final modified image x' :

$$x' = (1.0 - \alpha) \cdot x + \alpha \cdot \beta \cdot f(x) \quad (3.1)$$

where x is the original image and $\beta \cdot f(x)$ is the image modified by the β -regulated filter f .

In the rest of the thesis, to simplify the notation and increase readability, we will use the same term f to denote the parameterized version of the filter f .

In Table 3.1 we illustrate the results for the filter *Juno* when varying both the values of α and β simultaneously. We refer the reader to Appendix A for more examples of single filter applications.

3.3 The adversarial algorithm


























We generate the adversarial image x^* by applying to the clean image, x , a sequence of L optimized filters:

$$x^* = f_{k_1} \circ f_{k_2} \circ \dots \circ f_{k_L}(x) \quad (3.2)$$

where each f_i is the parameterized version of a filter selected from a set of predefined filters $F = \{f_1, f_2, \dots, f_F\}$.

We find the optimal sequence of L filters f_i and the values of their parameters (α_i, β_i) with a nested-evolutionary algorithm that consists of two components: an outer and an inner optimization. Specifically, the outer optimization step determines the sequence of

Table 3.1: Effects of filters with different α and intensity β values for Juno.

Juno	$\beta = 0.5$	$\beta = 0.75$	$\beta = 1.0$	$\beta = 1.25$	$\beta = 1.5$
$\alpha = 0.2$					
$\alpha = 0.4$					
$\alpha = 0.6$					
$\alpha = 0.8$					
$\alpha = 1.0$					

L filters $f_i \in F$ to use, while the inner optimization determines the best values of the corresponding parameters (α_i, β_i) . We choose a nested optimization approach in order to fine-tune the parameters of the filters and increase the performance of the adversarial method. Preliminary experiments (Table B.1) showed that applying the filters with default values leads to a less successful attack since the default modification does not result in an adversarial perturbation. Thus, the inner optimization aims to boost the adversarial power of the filters. The outer optimization is implemented by a Genetic Algorithm (GA), while the inner one is implemented by different strategies such as Genetic Algorithm, $(1, \lambda)$ -Evolution Strategies, Differential Evolution (DE), and a random-based approach with tournament (Rand-T).

We report the pseudo-code for the core adversarial algorithm in Algorithm 6 and provide

details on the algorithm components in Sections 3.3.1, 3.3.2.

3.3.1 Outer-optimization step

For the outer-optimization step, we employ a genetic algorithm to iteratively evolve a population of N_{out} randomly initialized candidate solutions towards an optimal solution over G_{out} generations. For each generation, we breed a new population of solutions by applying crossover on randomly selected population members followed by mutation. Each candidate solution is evaluated based on a task-specific objective function and only the best-performing ones are passed on to the next generation. The main algorithmic steps specific to our problem are summarized as follows:

1. **Initialization:** we generate the initial population by creating a set of N_{out} sequences with L randomly selected filters parameterized by $\alpha = 1$ and $\beta = 1$.
2. **Crossover:** we use a standard one-point crossover to create a new candidate solution (offspring) from two randomly selected individuals of the current population. For example, given two parents:

$$p = (f_{t_1}, f_{t_2} \dots, f_{t_L}) \tag{3.3}$$

and

$$q = (f_{k_1}, f_{k_2} \dots, f_{k_L}) \tag{3.4}$$

and a random crossover index $c \in \{1, \dots, L\}$, we obtain the offspring:

$$o_{p,q} = (f_{t_1}, \dots, f_{t_c}, f_{k_{c+1}}, \dots, f_{k_L}). \tag{3.5}$$

This guarantees that each offspring inherits genetic information from both parents, including their optimized parameters.

3. **Mutation:** based on a probability of mutation, we replace a filter f_j from an individual p with another randomly selected filter f_m from F . The substituent filter f_m has its parameters set to random values in order to perform mutation also over the parameters and to maintain genetic diversity from one generation to another. For instance, considering the individual $p = (f_{t_1}, f_{t_2} \dots, f_{t_L})$ and supposing that its filter f_{t_2} has been

Algorithm 6 General structure of the algorithm for finding the adversarial perturbation.

Require: Image x , target model M , outer population size N_{out} , outer generations G_{out} ,

inner population size N_{in} , inner generations G_{in}

- 1: Initialize population P of N_{out} individuals
- 2: **for** $p \in P$ **do**
- 3: generate x' using p ▷ Equation 3.2
- 4: evaluate the fitness \mathcal{F} of p by querying M on x'
- 5: **end for**
- 6: **for** $g \in G_{out}$ **do**
- 7: Offspring = \emptyset
- 8: **for** $i \in N_{out}$ **do**
- 9: select randomly two parents p_1, p_2 from P
- 10: $o_{p_1, p_2} \leftarrow \mathbf{crossover}(p_1, p_2)$
- 11: $\bar{o} \leftarrow \mathbf{mutation}(o_{p_1, p_2})$
- 12: $o \leftarrow \mathit{Optimizer}_{params}(\bar{o}, G_{in}, N_{in})$
- 13: Add o to Offspring
- 14: **end for**
- 15: **for each** $o \in \text{Offspring}$ **do**
- 16: generate x' using o ▷ Equation 3.2
- 17: evaluate the fitness \mathcal{F} of o by querying M on x'
- 18: **end for**
- 19: $P = \mathbf{selection}(P, \text{Offspring})$
- 20: **end for**
- 21: **return** p_{best} best sequence of filters from P

chosen to mutate with a filter $f_m \in F$, the new mutated individual \bar{p} is:

$$\bar{p} = (f_{t_1}, f_m, f_{t_3}, \dots, f_{t_L}) \quad (3.6)$$

where the values of parameters α_m and β_m are randomly extracted from the respective parameter domains.

4. **Evaluation:** In order to evaluate a population member we have to: (i) build the corresponding modified image x' using (3.2) and (ii) determine how effective and desirable the modified x' is, according to the goals (attack success rate, image quality, etc.) we want to reach for the specific task (image classification, object detection, etc.). The evaluation functions may include calls to the target model, image quality assessment, calls to attack detection methods, and so on. For example, if we include in the objectives the attack Success Rate, or any other model performance measure, we have to query the model M in order to determine how effective the attack is by comparing the output of the perturbed image $M(x')$ with the output of the original image $M(x)$.
5. **Selection:** at the end of each generation, after the evaluation of candidate solutions, we select the N_{out} fittest individuals from the set of $2N_{out}$ candidates composed of the current population members and their offspring. The fitness function can be formulated either as a single-objective or as a multi-objective function. Thus, based on the scenario, we adopt different selection strategies.

In the single-objective context, we rank the individuals according to their fitness value and select the top N_{out} individuals (Algorithm 7).

In the case of multi-objectives, two conflicting fitness functions are computed for each individual (parents and offspring). Therefore, we use the *non-dominated sorting* and *crowding distance* procedures of the NSGA-II [83] algorithm for multi-objective selection to obtain the Pareto front, which is defined as the set of non-dominated solutions, where each objective is considered as equally good. At the end of the optimization process, we select the point closest to zero as the final best solution.

Algorithm 7 Selection procedure for single objective fitness functions.

Require: Current population P , Offsprings

- 1: $S \leftarrow \text{Ranking_decreasing_by_fitness}(P, \text{Offsprings})$
 - 2: **return** $S[0 : N_{out}]$ // N_{out} is the size of P
-

Algorithm 8 Selection procedure for multi-objective fitness function

Require: Current population P , Offsprings

- 1: $S \leftarrow \text{Non_dominated_sorting}(P, \text{Offsprings})$ ▷ Algorithm 3
 - 2: $S \leftarrow \text{Crowding_distance_sorting}(S)$ ▷ Algorithm 5
 - 3: **return** $S[0 : N_{out}]$ // N_{out} is the size of P
-

3.3.2 Inner-optimization step

The inner optimizer is called for each element after the mutation operation (Algorithm 6, line 12). The task of the inner optimizer is to evolve a population of N_{in} lists of parameters (α_i, β_i) of each mutated individual \bar{o} to determine the values that improve the fitness of the element. The sequence of filters in \bar{o} remains fixed during the inner optimization and only the parameters are updated. By doing this, we aim at further exploring the space of the parameters' values and finding the best set of parameters for each element returned by the outer optimization step. Both outer and inner optimization steps use the same fitness function to evaluate population members.

We explore different strategies for the inner optimization: a Genetic Algorithm (GA), $(1, \lambda)$ -Evolutionary Strategy, Differential Evolution (DE), and a random-based approach with tournament (Rand-T). For GA, ES, and DE we follow the structure of the algorithms described in Sections 1.2.1, 1.2.3, 1.2.2. In the random-based approach (Rand-T) we skip the inner optimization step and we change the parameter values randomly. Specifically, given a mutated element \bar{o} , we generate o by randomly changing the filters' parameters of \bar{o} . In this case the selection process is implemented as a 2-way tournament between \bar{o} and o where they compete against each other and only the fittest candidate is passed on to the next generation. We use this Rand-T implementation as the baseline for the experiments.

3.3.3 Queries to the target model

The proposed attack works with limited access to the target model. The target model is queried every time we have to compute the objective function. Besides the explicit calls to the evaluation function made in the initialization step and in the offspring evaluation step of the outer iteration, we have to consider the $G_{in} \times N_{in}$ calls made by the inner optimization phase for each member in the outer population. The maximum number of queries can be computed as:

$$Q_{max} = N_{out} + G_{out} \times (N_{out} \times G_{in} \times N_{in} + N_{out}) \quad (3.7)$$

Reducing the number of queries to the target model is crucial for two reasons: it reduces the computational cost, and it lowers the risk of being detected and banned by the target model during an attack.

Chapter 4

Attacks on image classifiers

In this chapter we validate our adversarial attack on image classification considering three different attack configurations: per-instance single attack (Section 4.1), per-instance multi-objective attack (Section 4.2), and universal multi-objective attack (Section 4.3).

4.1 Per-instance Single Objective Attack

In this section, we study the effectiveness of the proposed attack on state-of-the-art image classification models trained on a large scale visual recognition dataset [143]. We guide the optimization process using a success-based objective function (Equation 4.2) to generate image-dependent perturbations. We refer to this setup as *per-instance single objective* attack. We introduce our approach for this case study and empirically assess the performance of the adversarial attack.

We propose to generate adversarial examples by optimizing image filters that resemble those available on Instagram in a black-box setup. Differently from ACE [29], our method does not require full access to the target model and only uses the final output of the target model. Moreover, we combine multiple filters to craft more robust and transferable perturbations and generate images with more diverse visual effects, with styles ranging from subtle and warm looks to more dramatic and vivid colors, as shown in Figure 4.1. By simulating the effects of Instagram filters and by targeting specific image attributes we aim to reduce human awareness towards the applied modification without using additional resources that can



Figure 4.1: Adversarial images generated on MobilenetV1 (top row), VGG19 (middle row), and ResNet50 (bottom row) with our method with 3, 4, 5, and 6 filters and ColorFool (CF).

hinder the image quality and add extra computational cost to the adversarial optimization process (i.e. image segmentation models [31]).

4.1.1 Problem formulation

Let x be a benign RGB image. Let M be an image classifier that predicts the class label for a given input image x , that is $M(x) = y$. We formulate the problem of finding an adversarial example x^* , a perturbed version of x that fools the classifier to produce a classification label different from that of the original image, as the following optimization problem:

$$x^* = \operatorname{argmax}_{x'} \mathcal{F}(x, x') \quad (4.1)$$

with

$$\mathcal{F}(x, x') = \begin{cases} 1, & \text{if } M(x) \neq M(x') \\ 0, & \text{otherwise} \end{cases} \quad (4.2)$$

where x' is obtained by applying a sequence of parameterized filters to x (Equation 3.2) and x^* represents the best solution among all perturbed versions of x .

4.1.2 Experimental setup

Target Networks: We validate the proposed attack on three state-of-the-art pre-trained and publicly available image classifiers: MobileNet-v1 (M-v1) [144], ResNet50 (R50) [145], and VGG19 [146]. We choose models with different architectures in order to explore the generalization ability of the algorithm and to study the transferability degree of adversarial examples. Moreover, these models are often fundamental components of other deep learning systems used for object detection or image segmentation. Thus, analyzing the robustness of these networks can provide insight into the vulnerability of more complex systems. Finally, the choice of MobileNet is motivated because it is often used in machine-learning-powered mobile apps, thanks to its low latency and lightweight nature, which makes them an interesting case study.

Dataset: We use the ImageNet [143] classification dataset. We sample randomly 1 image for each of the 1000 classes from the validation dataset and we preprocess the images according to the requirements of the pre-trained neural networks.

Attack configuration: We use the following hyperparameters for the outer optimization set: population size $N_{out} = 10$, number of generations $G_{out} = 10$, and mutation probability $\rho = 0.5$. In order to choose the algorithm for the inner optimization step we performed some preliminary experiments using both Evolution Strategies (ES) and Differential Evolution (DE). We configured ES with a population size $N_{in} = 5$ and a number of generation $G_{in} = 3$, an initial learning rate = 0.1, and a decay rate of 0.75. In Figure 4.2 we report the success rate over the number of generations when using ES as inner optimizer. In general, the above configuration with 10 outer generations is sufficient to achieve a high success rate.

We compare our method against ColorFool [31], a black-box state-of-the-art method similar to our attack that uses unrestricted color manipulation to generate adversarial examples. ColorFool only requires the number of iterations to be set. However, it requires an image

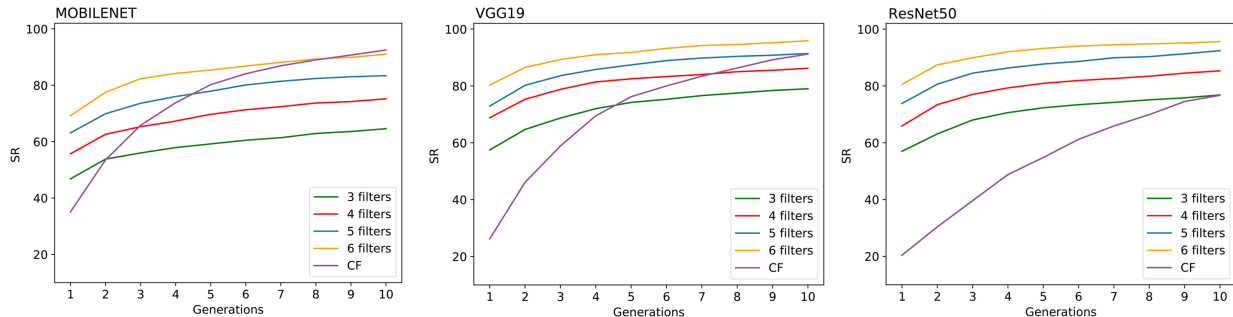


Figure 4.2: Success Rate (SR%) w.r.t number of generations for MobileNet, VGG19, and ResNet50 with our method and ColorFool(CF). In each generation 160 queries are made to the target model.

segmentation model to identify the semantic regions in the image which leads to extra computational cost and running time. Moreover, ColorFool also involves a human intervention to manually select the sensitive semantic classes, such as sky, water, plants and humans. In our analysis we decided to focus only on the part involving the generation of the adversarial image, ignoring the additional running time generated by the segmentation model.

Moreover, we observed that DE is producing slightly better results in terms of success rate, but it is $\approx 3.5 \times$ slower than ES. Thus, we selected ES as inner optimizer to align the computational time of our method with ColorFool [31] and ensure a fair comparison under the same experimental conditions.

4.1.3 Evaluation and Experimental Results

We evaluate the effectiveness of the attack in terms of success rate and transferability rate. Let M be the target model, we define the *Success Rate* (SR) as:

$$SR(X, X^*) = \frac{1}{|X|} \sum_{i=1}^{|X|} M(x_i) \neq M(x_i^*), \quad (4.3)$$

where x_i is the i -th image of the original dataset X , and x_i^* is the corresponding perturbed image in the set of all perturbed images X^* .

Since transferability represents the ability of the adversarial examples crafted for a particular model to mislead other different unseen models, we measure the *Transferability Rate*

(*TR*) as:

$$TR(\bar{X}^*) = \frac{1}{|\bar{X}^*|} \sum_{i=1}^{|\bar{X}^*|} N(x_i) \neq N(x_i^*), \quad (4.4)$$

where $\bar{X}^* \subset X^*$ is the set of successful adversarial examples generated on a model M and N is an unseen model $N \neq M$. The success rate and transferability rate are shown in Table 4.1.

An attack can also be evaluated considering its deception ability. We measure the deception ability of the attack using the following metrics: *Success Rate at rank k* (*SR@k*) [147], *Old Label New Rank* (*OLNR*) [148] and *New Label Old Rank* (*NLOR*) [148].

Given a clean image x , let *old label* $c_{old} = M(x)$ and *new label* $c_{new} = M(x^*)$ be the classes respectively predicted for the original and perturbed image by the model M , and $rank(c, x)$ the rank of the class c in the probability distribution returned by the model M for the image x . Then, we extend the definition of success rate to consider the new rank of the *old label* using *SR@k* defined as:

$$SR@k(X, X^*) = \frac{1}{|X|} \sum_{i=1}^{|X|} rank_k(x_i, x_i^*) \quad (4.5)$$

$$rank_k(x_i, x_i^*) = \begin{cases} 1, & \text{if } rank(c_{old}, x_i^*) > k \\ 0, & \text{otherwise} \end{cases} \quad (4.6)$$

with x_i, x_i^*, X, X^* as previously defined. The attack is considered successful only if it assigns to the *old label* c_{old} a rank greater than k after the attack, where k is a user-specified value.

Considering the notations proposed in [148] we obtain that $OLNR = rank(c_{old}, x^*)$ and $NLOR = rank(c_{new}, x)$. A strong attack will have high *NLOR*, meaning that the prediction changed to a label that had a low rank before the attack, and high *OLNR* meaning that the probability of the original class for the perturbed image x^* is small. We computed *NLOR* and *OLNR* for all adversarial examples and reported the mean values in Table 4.2.

An adversarial attack should also be reliable and robust to defense mechanisms. We propose to quantify the robustness as the percentage of adversarial examples that remain adversarial with respect to the original label prediction after passing through a defense framework. Since our attack uses image filters to generate the adversarial perturbation, we use a

Table 4.1: Success Rate (SR%, in gray cells) and Transferability Rate (TR% in white cells) against MobileNet-v1 (M-v1), VGG19 and ResNet50 (R50). The higher the SR (TR), the most successful (transferable) the attack. AC indicates the classifier under attack, TC indicates the classifier used for the transferability test. CF stands for ColorFool [31].

Attack	TC \ AC	M-v1	VGG19	R50
	AC			
Ours 3f	M-v1	64.60	51.70	52.94
	VGG19	34.81	79.00	50.89
	R50	34.24	48.31	76.80
Ours 4f	M-v1	75.20	53.65	60.51
	VGG19	39.68	86.20	53.13
	R50	38.92	56.98	85.30
Ours 5f	M-v1	83.50	63.55	65.83
	VGG19	45.08	91.40	59.41
	R50	45.56	61.36	95.60
Ours 6f	M-v1	90.00	71.00	71.00
	VGG19	49.11	95.90	65.48
	R50	46.86	67.26	95.60
CF	M-v1	92.30	21.50	11.30
	VGG19	36.30	91.10	12.30
	R50	48.60	44.30	76.70

defiltering framework to mitigate the visual effects of the applied image filters. Specifically, we choose Instagram Filter Removal Net [73] (IFR-Net, see Section 1.1.2).

Thus, given a set of adversarial images \bar{X}^* generated on a target model M , we measure the robustness of the attack as:

$$R(\bar{X}^*) = \frac{1}{|\bar{X}^*|} \sum_{i=1}^{|\bar{X}^*|} M(D(x_i^*)) \neq M(x_i) \quad (4.7)$$

where D is the defiltering model represented by IFR-Net and returns the defiltered version

of x_i^* .

Success Rate: We report in Table 4.1 the success rate (in gray cells) and transferability rate (in white cells) obtained by Colorfool and our technique with different numbers of filters, obtained attacking different networks (M-v1, VGG19, and R50). Both adversarial methods reach high SR against the direct target model. When using at least 5 filters, our method achieves higher SR than ColorFool on VGG19 (95.9% vs. 91.1%) and ResNet50 (95.6% vs 76.7%), and similar performance on MobiletNet-v1 (90.00% vs 92.3%). We notice that ResNet50 is quite robust against ColorFool attack but very vulnerable to our filter-based perturbations: we obtain higher SR even when using 3 or 4 filters. On the other hand, our method is not as powerful on MobileNet-v1 as ColorFool, requiring 6 filters to reach competitive results and achieving the lowest SR across all methods and networks. Nonetheless, we observe that the proposed method exhibits the highest transferability rate, with an average TR of 46.11% versus 29.05% obtained by ColorFool. When using ColorFool, ResNet50 confirms itself the most robust also from the transferability point of view, since only 12.3% of adversarial examples from VGG19 succeed to attack it. The adversarial examples crafted with our method achieve much higher TR even when using only 3 filters, ranging from 34.24% to 71.00% overall. Thus, when performing attacks by transferability, our method is preferable.

Deception ability: We evaluate the deception ability of our method and ColorFool using $SR@k$, $OLNR$, and $NLOR$. For $SR@k$, we choose $k = 5$. In this case, an attack is considered valid only if the label of the original image is not in the Top-5 predictions after the adversarial manipulation. We show the mean and standard deviation of $OLNR$ and $NLOR$, and $SR@5$ in Table 4.2. The results indicate that our method significantly outperforms ColorFool on every metric. MobileNet-v1 is the hardest to deceive, confirming again its robustness while the higher deception is obtained on ResNet50. Interestingly, although ResNet50 was the most robust to ColorFool in terms of success rate we observe the highest $NLOR$ among all methods under the ColorFool attack. More surprising is the fact that ColorFool, despite achieving some of the highest SR , achieves a maximum $SR@5$ of 1.2%. Instead, our method has $SR@5$ ranging from 9.7% to 40.1%.

Table 4.2: Evaluation of Deception ability with *OLNR*, *NLOR* and *SR@k*. Higher values indicate stronger attacks. Our method outperforms ColorFool (CF) on every metric. Note that ColorFool has $SR@5 \approx 1\%$.

Attack	Model	<i>OLNR</i>	<i>NLOR</i>	<i>SR@5</i> %
Ours 3f	M-v1	5.87 ± 20.80	13.35 ± 50.43	9.70
	VGG19	7.13 ± 18.81	21.96 ± 67.82	15.90
	R50	12.40 ± 50.02	17.40 ± 50.28	17.60
Ours 4f	M-v1	7.50 ± 26.79	20.28 ± 64.93	14.80
	VGG19	15.92 ± 53.00	33.29 ± 91.60	24.50
	R50	14.65 ± 63.77	27.99 ± 78.36	22.30
Ours 5f	M-v1	16.54 ± 68.46	33.29 ± 95.30	22.50
	VGG19	20.95 ± 73.46	38.55 ± 100.93	32.60
	R50	26.99 ± 92.48	46.18 ± 108.50	33.30
Ours 6f	M-v1	21.76 ± 81.36	46.75 ± 122.40	31.20
	VGG19	27.73 ± 91.78	49.53 ± 116.45	37.80
	R50	36.41 ± 111.99	71.36 ± 157.63	40.10
CF	M-v1	2.16 ± 0.60	14.14 ± 48.40	0.60
	VGG19	2.50 ± 0.80	14.06 ± 42.43	1.20
	R50	2.27 ± 1.182	60.39 ± 171.56	1.10

Table 4.3: Robustness (R%) of adversarial attacks. Higher values indicate more robust attacks. Adversarial images generated with our method are more robust against the IFR-Net image filtering defense than examples generated with ColorFool (CF).

Attack \ Model	Our 3f	Our 4f	Our 5f	Our 6f	CF
M-v1	76.00	63.82	71.11	70.55	46.21
VGG19	66.70	72.16	72.65	73.93	46.04
R50	65.23	68.34	71.32	72.70	36.63

Robustness: To compute the robustness of attacks we apply IFRNet to all successful adversarial examples and then check how many of them remain adversarial with respect to the classification label of the original image. We report the robustness scores in Table 4.3. We find IFRNet to be more effective on ColorFool than on our method which explicitly tries to imitate the visual effects of Instagram filters. Despite the fact that IFRNet is able to remove the effects of single image filters to a great extent [73], it cannot neutralize the adversarial perturbations obtained by mixing multiple filters. Figure 4.3 shows adversarial examples before and after the IFRNet application.

Image quality: We assess the image quality of the adversarial examples using NIMA [64] and NIQE [65]. Both NIMA scores (Table 4.4) and NIQE scores (Table 4.5) show that there is no significant difference between the quality of the adversarial examples and the quality of the original images. Moreover, our attack is able to produce adversarial examples with image quality equivalent to ColorFool without specifically defining the color ranges of each semantic region which is more computationally convenient.

Query Efficiency: In a black-box setup query efficiency is considered a key characteristic for generating realistic attacks. Limits on the number of queries can arise from time or budget constraints when querying incurs costs, as noted in [30]. Considering the experimental configuration, our system allows a maximum of 1610 queries to the victim model. We report the results obtained on the ImageNet dataset by systems that can be considered query-

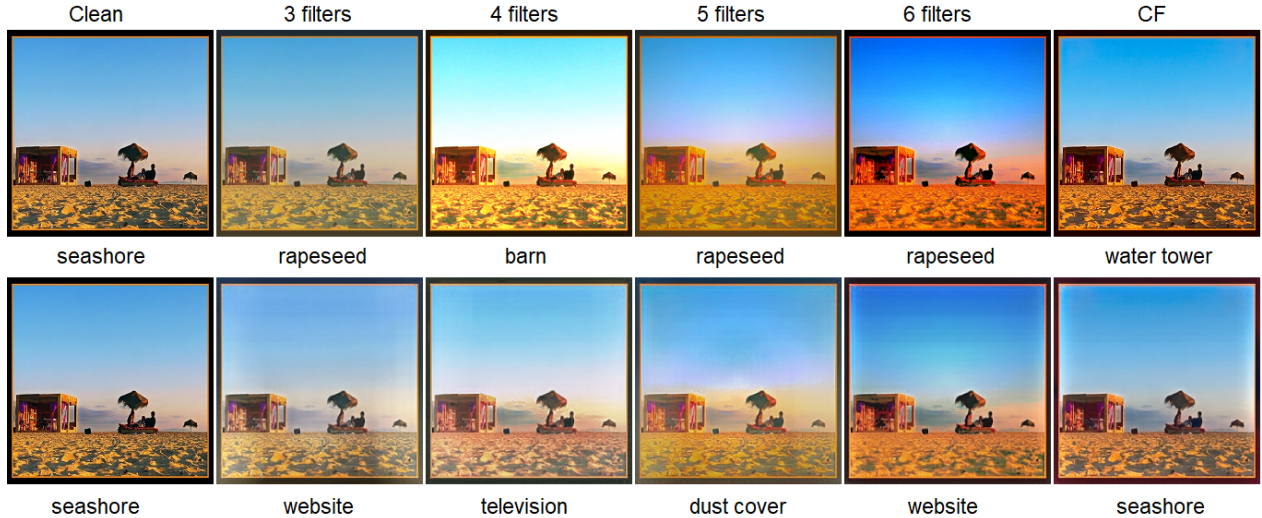


Figure 4.3: Adversarial images generated on VGG19 before IFRNet (top row) and after de-filtering with IFRNet (bottom row). The adversarial examples generated with our method remain adversarial even after the application of IFRNet, while the one generated with ColorFool (CF) is reverted to the original label.

Table 4.4: Image quality with NIMA (the higher, the better) of the adversarial images generated against MobileNet-v1 (M-v1), VGG19, ResNet50 (R50). In each cell x/y : x is the mean NIMA score on the adversarial images, and y is the mean NIMA score on the corresponding clean images.

Attack \ Model	Our 3f	Our 4f	Our 5f	Our 6f	CF
M-v1	5.15/5.17	5.15/5.17	5.14/5.18	5.13/5.18	5.18/5.18
VGG19	5.15/5.18	5.14/5.18	5.13/5.18	5.14/5.19	5.19/5.18
R50	5.14/5.17	5.15/5.18	5.13/5.18	5.13/5.18	5.20/5.18

efficient methods working in a label-only black-box setting: in [30], that is based on an evolutionary strategy, the attack needs 270k queries to reach 90% success rate; in [56] the authors report that their best system can produce attacks with good quality in the 76% of cases with 10k queries or in the 98% of cases with 20k queries; In [34] the authors report that they can reach the 50% of accuracy with 30k queries, but they need 160k queries to

Table 4.5: Image quality with NIQE (the lower, the better) of the adversarial images generated against MobileNet-v1 (M-v1), VGG19, ResNet50 (R50). In each cell x/y : x is the mean NIQE score on the adversarial images, and y is the mean NIQE score on the corresponding clean images.

Attack \ Model	Our 3f	Our 4f	Our 5f	Our 6f	CF
M-v1	31.65/32.70	31.21/32.74	31.14/32.78	30.05/32.58	32.03/32.39
VGG19	31.66/32.61	31.32/32.44	30.39/32.70	29.98/32.52	32.04/32.37
R50	32.08/32.77	31.14/32.53	30.61/32.56	29.94/32.54	31.63/32.40

reach the 90%. ColorFool is also query-efficient, achieving a high attack success rate within a limited query budget.

Our method, with the same query budget, achieves a comparable success rate and image quality, but offers more transferable, robust, and deceptive attacks. It also requires less iterations in order to achieve high attack success rate and to reach convergence (Figure 4.2). The reason for the difference in behaviour and adversarial characteristics relies in the way the two methods modify the images. In the case of ColorFool, all pixels in a semantic region are perturbed with the same intensity which limits the variability of pixels and reduces the possibility of finding an adversarial combination of pixel values. Our method uses more complex image manipulation operations which does not disrupt the original color diversity and increases the pixel value variation since pixels from the same semantic region will be affected differently by the filter. The experimental results show that the optimized combination of image filters can successfully fool state-of-the-art image classifiers. We hope that our findings will encourage more research and studies on the vulnerability of such models against image filtering based attacks.

4.2 Per-instance Multi-objective Attack: Emotion Recognition

Visual *Emotion Recognition* (ER) is one of the first *Affective Computing* techniques [149] that have been widely studied in computer science and artificial intelligence, based on visual features of facial expression. Deep learning approaches for facial emotion recognition obtain high accuracy on basic emotion models, e.g., Ekman’s models [150], in the specific domain of facial emotional expressions. Thus, facial tracking of users’ emotions could be easily used against the right to privacy or for manipulative purposes. For instance, in behavior-tracking applications, the emotional reactions of a user in front of a product could produce extremely precious insights for companies, governments, or political parties, prying into the user’s habits and emotional states. E.g., marketing applications in a supermarket, in front of a shop showcase, or browsing an e-commerce website [151, 152]; tracking drivers’ states [153]; analyzing pieces of information in social networks [154]; analyzing news or political opinions [155]; military robot interaction [156]. Due to the critical nature of such information, tracking it could open a breach in personal data confidentiality, and become a potential source of manipulation bias for the user’s preferences. Thus, the diffusion and the wide use of deep learning-based systems pose significant security and privacy issues.

To guarantee the user’s freedom and defense against emotion recognizers in settings where they may be unauthorized, we suggest using our adversarial technique to filter out the emotional features from video frames and photographs of human faces. By perturbing the images with adversarial filters the information extraction process becomes more difficult, thus aiming for privacy protection. In this context, we extend the *per-instance single objective* approach described in Section 4.1 to a *per-instance multi-objective* that allows us to account for a second objective when crafting the adversarial images. Specifically, we consider an image quality assessment metric (i.e SSIM [63]) to prevent unnatural excessive alteration and control the level of applied perturbations.

4.2.1 Problem formulation

Differently from the attacks to image classifiers that work with a wide range of types of images (i.e. ImageNet dataset) and where heavier modification can be accepted (i.e as artistic stylization and/or personal preference), in the task of emotion recognition it is particularly important to avoid perturbing the images (i.e. faces) excessively and ensure that the adversarial image maintains a natural-looking aspect. Thus, we formulate the black-box attack as a Multi-Objective Optimization Problem to combine the *per-instance* approach presented in Section 4.1 with a full-reference image quality assessment metric to control the amount of adversarial perturbations applied to the images.

Given M a target *facial emotion recognizer*, x an original facial image, x' the perturbed image, we define the multi-objective problem of our interest as:

$$x^* = \underset{x'}{\operatorname{argmin}} \mathcal{F}_{MO}(x, x') \quad (4.8)$$

with

$$\mathcal{F}_{MO}(x, x') = \{1.0 - \mathcal{F}(x, x'), 1 - SSIM(x, x')\} \quad (4.9)$$

where \mathcal{F} is the adversarial attack indicator function introduced in (4.2) and $SSIM$ is an automatic full-reference perceptual metric introduced by Wang et al. [63] that quantifies the image degradation as perceived changes in the structural information. We found that optimizing with respect to this metric reduces the presence of artifacts in the images.

4.2.2 Experimental setup

Emotion Recognizer: We evaluate the proposed attack on an emotion recognition (ER) neural network designed to classify a human face in the seven basic emotions of the Ekman’s model [157]: *Anger, Contempt, Disgust, Fear, Happiness, Sadness, Surprise*, extended with an eighth *Neutral* class. We use transfer learning to adapt the MobilNetV2 [144] pre-trained on the ImageNet dataset for the emotion recognition task. Specifically, we replace the last MobileNetV2 fully connected classification layer with a fully connected layer of size 128, dropout of 0.5 and ReLU activation functions followed up with a final fully connected layer of size 8 for the emotion classes and Softmax activation. We fine-tune the

ER model on the AffectNet [158] dataset using the Stochastic Gradient Descent with momentum (SGDM) optimizer with a batch size of 10 and an initial learning rate of $1e - 3$ which is decayed by a factor of 0.5 every 10 epochs on a total of 80. Cross-validation was used for the hyperparameters tuning. The trained emotion recognition model is available at <https://github.com/Ellyuca/AGV-Project>.

Dataset: The AffectNet [158] dataset is one of the most used datasets for Emotion Recognition. It contains ≈ 261 K images labeled within the eight categories of the extended Ekman model with Happiness and Neutral classes accounting together for about 2/3 of the dataset. Thus, we sample randomly 3.5K images per class to obtain a perfectly balanced dataset, for a total of 28K images. We split the dataset in training and testing using a stratified sampling strategy with a ratio of 80%-20%. We perform data augmentation by applying random horizontal flipping and horizontal/vertical shifting to the images, by a random offset in the $[-15,+15]$ pixels range. To test its robustness against our adversarial attack, we select 10 correctly classified images for each class from the testing set, for a total of 80 images.

Attack implementation details: Based on the finding from the previous case study, we set the following parameters for the optimization algorithm: the population size of the outer is $N_{out} = 10$, the mutation probability $\rho = 0.5$, and number of generations $G_{out} = 10$. For the inner optimization we use Evolutionary Strategy (denoted as Inner-ES) and Differential Evolution (denoted as Inner-DE). We configure the Evolutionary Strategy with a population size of $\lambda = 5$ and number of generations $G_{in} = 3$, an initial learning rate = 0.1 and decay rate = 0.75. For Differential Evolution we choose a *rand/1/bin* strategy with a population size = 5, number of generations =3, F=0.7 and CR=0.5. The parameters α and β of each filter are initialized with default values equal to 1. We run experiments with 3, 4, and 5 filters to show the ability of the attack, formulated as a multi-objective optimization, to generate adversarial examples. Figure 4.4 shows that the method is able to converge towards high success rates under the above configuration.

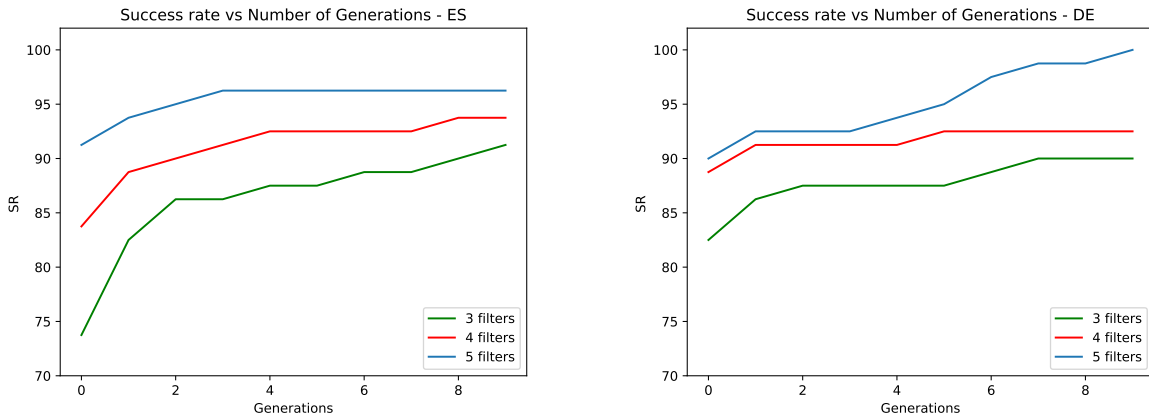


Figure 4.4: Success rate over the number of generations. A number of generations = 10 is sufficient to achieve high success rates for both ES (left) and DE (right).

Table 4.6: Success Rate ($SR\%$) with ES and DE when using 3,4, and 5 filters.

Optimization	3 filters	4 filters	5 filters
Inner-ES	91.25	93.75	96.25
Inner-DE	90.00	92.50	100.00

4.2.3 Evaluation and Experimental Results

We measure the success rate (SR) of the adversarial attack with as the ratio between the number of images for which the emotion recognizer fails the classification and the total number of images used for the attack. We use the metric defined in 4.3.

We also compute the success rate at different SSIM threshold values (SR_t) to analyse the effectiveness of the attack with respect to the amount of the adversarial perturbation computed by means of SSIM index. We define SR_t as follows:

$$SR_t(X, X^*) = \frac{1}{|X|} \sum_{i=1}^{|X|} A(x_i, x_i^*), \quad (4.10)$$

with

$$A(x_i, x_i^*) = \begin{cases} 1, & \text{if } M(x_i) \neq M(x_i^*) \wedge SSIM(x_i, x_i^*) \geq t \\ 0, & \text{otherwise} \end{cases} \quad (4.11)$$

where $t \in \{0.05, \dots, 0.95\}$ with a step of 0.05. The results are presented in Figure 4.8.

The algorithm has been tested both for ES and DE as methods for the parameter optimization phase (inner phase in the Algorithm 6). The experimental results (Table 4.6) show that the algorithm can reach a high attack success rate (SR). More specifically, it achieves 91.25%, 93.75%, and 96.25% when using ES as inner optimization and 90.00%, 92.50%, and 100.00% when using DE, both with 3, 4, and 5 filters, respectively.

Moreover, in Figures 4.5-4.6-4.7 we report the error distribution among classes using the confusion matrices obtained from the three experiments. We note that an increase of the length of filter sequences corresponds to increasing SR and that the only classes that maintain some correct classifications are *Fear* and *Happiness* while all the others show an SR of 100%. Moreover, we observe that for the classes *Contempt*, *Neutral* and *Surprise* we obtained a shift (a number of errors greater than 50%) towards another class (*Contempt* \rightarrow *Happiness*, *Neutral* \rightarrow *Sadness* and *Surprise* \rightarrow *Fear*), while for the other 5 classes, the errors are quite-uniformly distributed among the other classes. The analysis indicates that the emotions of *Fear* and *Happiness* are the most resilient to attacks and easier to be recognized by the model. This could be due to the fact that, in general, these emotions are characterized by strong visual features which offer higher discriminative power that help the model to better identify the emotions. Thus, in order to attack such emotions heavier image manipulations are required, as shown by the our experiments: increasing the number of filters leads to an increment of classification errors for *Fear* and *Happiness*. On the other side, emotions such as *Contempt* and *Neural* are more difficult to recognize by the model since they might share similar visual attributes which increase the degree of confusion regarding these emotions and leaves the model more prone to attacks.

In Table 4.7 we show the visual effects produced by the adversarial filters on the images. For each original image in the first column, the results obtained for sequences of 3, 4, and 5 filters are reported. The algorithm generates natural-looking and artifacts free adversarial images. This effect is due to the uniform application of the filters across the entire image, and the controlled perturbations through the SSIM index. Moreover, to have a global view of the impact in terms of SSIM index, we have analyzed the values of the index for all the attacking images. In Figure 4.9 the distributions of the SSIM values for the images produced by sequences of 3, 4, and 5 filters are shown. For most of the images, the scores



Figure 4.5: Confusion matrix for the attack with 3 filters with ES (left) and DE (right).

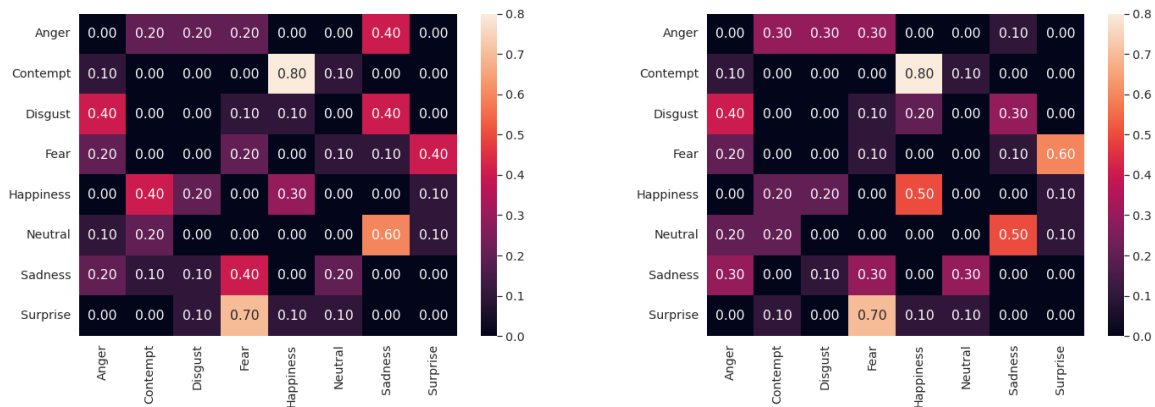


Figure 4.6: Confusion matrix for the attack with 4 filters with ES (left) and DE (right).

are remarkably high (1 is the upper bound), and only for very few cases, they reach values under 0.7. Furthermore, there is no significant difference among the three versions: users can choose according to their necessities, preferring a less or more modified image at the expense of the effectiveness of the protection.

In Figure 4.8 we show the results obtained using the SR_t metric. We notice that high success rate values correspond to a bigger dissimilarity between the original and adversarial images and the SSIM values decrease as we increase the number of filters used for the attack. We could apply a constraint on the SSIM if having high similarity between images is of absolute necessity. For example, with a threshold of 0.8 on the SSIM the attack is still effective with success rate of 75.00%, 70.00%, 75.00% for 3, 4, and 5 filters respectively.

We analysed also the computational time of the algorithm. In Table 4.8 we report the average time in seconds necessary to complete one outer generation. *Inner-DE* achieves the

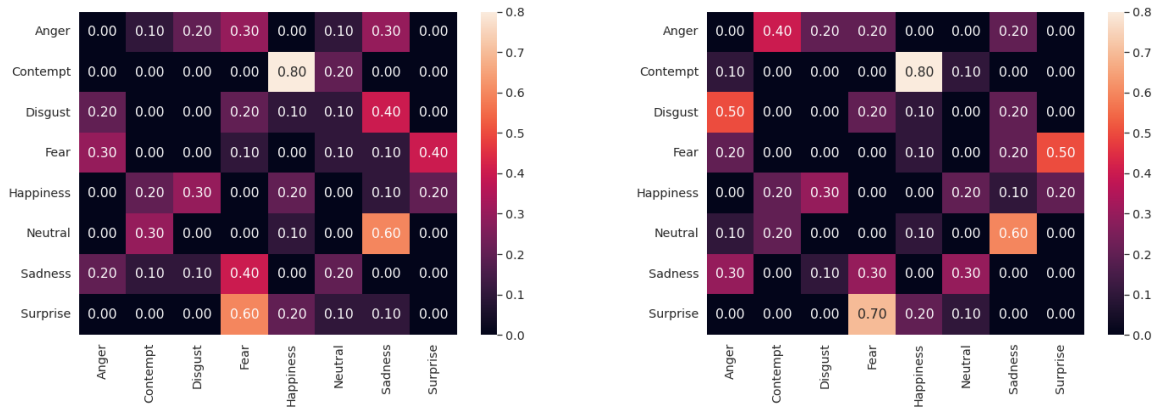


Figure 4.7: Confusion matrix for the attack with 5 filters with ES (left) and DE (right).

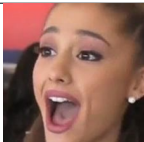
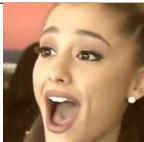
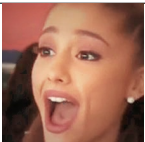
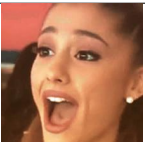
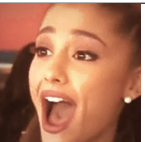
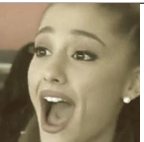
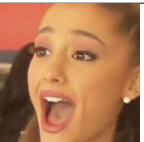

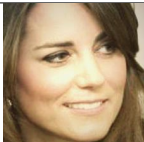

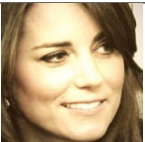
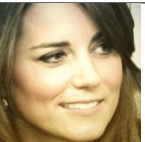





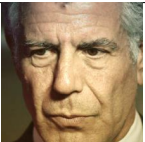

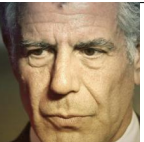
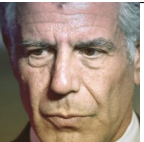







Original	3 filters		4 filters		5 filters	
	ES	DE	ES	DE	ES	DE
 surprise	 fear	 fear	 fear	 fear	 fear	 fear
 happiness	 contempt	 contempt	 disgust	 disgust	 contempt	 disgust
 anger	 sadness	 sadness	 sadness	 sadness	 sadness	 sadness
 happiness	 contempt	 contempt	 contempt	 contempt	 contempt	 contempt

Table 4.7: Examples of adversarial samples: the first column reports the original image and original classification. Columns 2-4 show the adversarial images with their classification. We can notice how the adversarial attack changes the automated emotion recognition without disrupting the image appearance.

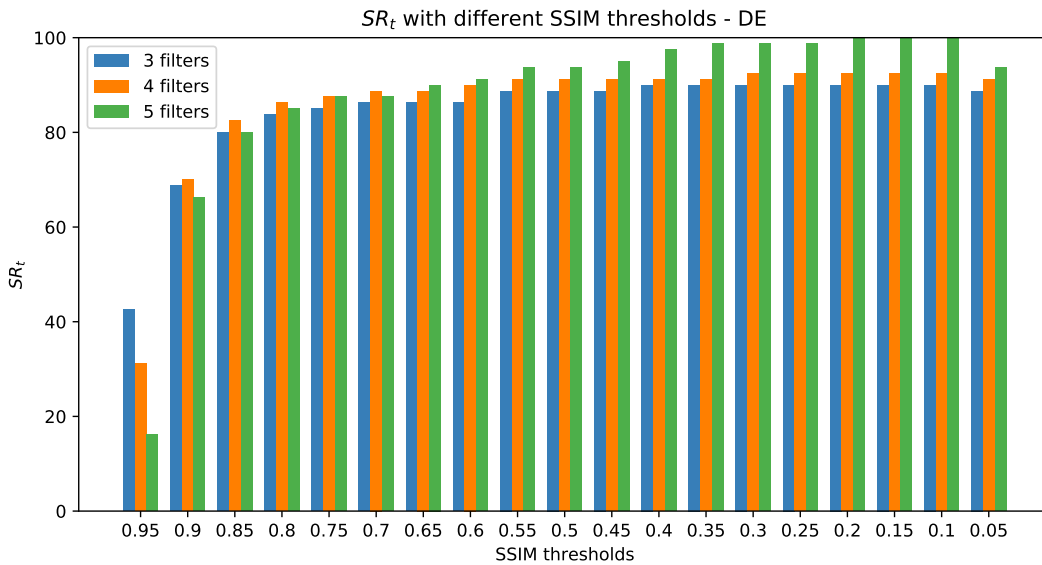
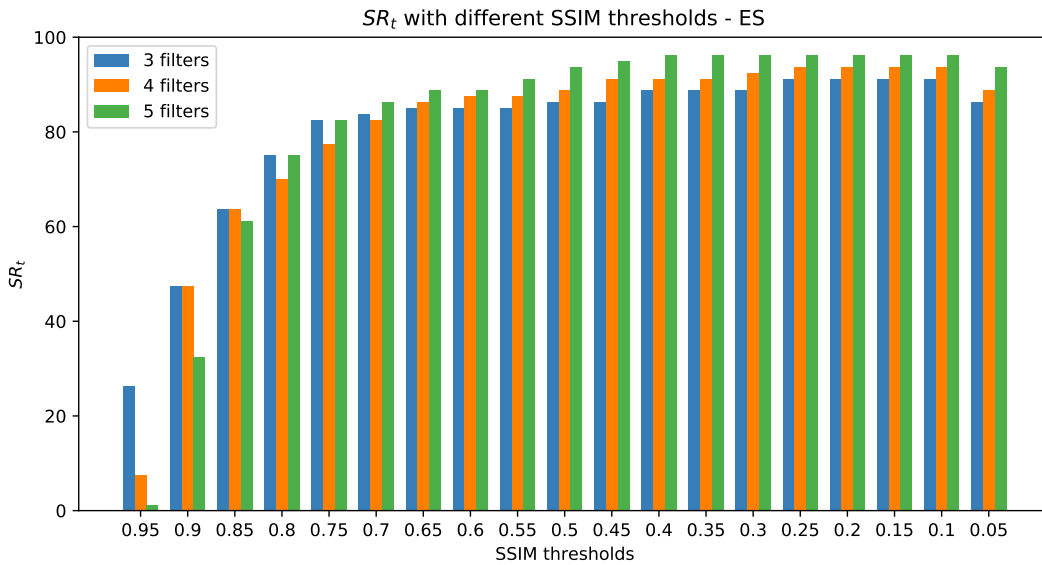


Figure 4.8: Success rate computed with different SSIM threshold values for the attack with ES (top) and DE (bottom) with 3, 4, and 5 filters, respectively.

best results but, from a computational time perspective, it is the most expensive, with an average of $\simeq 70$ seconds to complete one outer generation. On the other hand, an ablation study (see Appendix B.2) shows that simpler variants are more time efficient but achieve a lower success rate. *Inner-ES* has the best attack effectiveness-time trade-off, generating adversarial images with high similarity scores and being $3.5\times$ faster than *Inner-DE*.

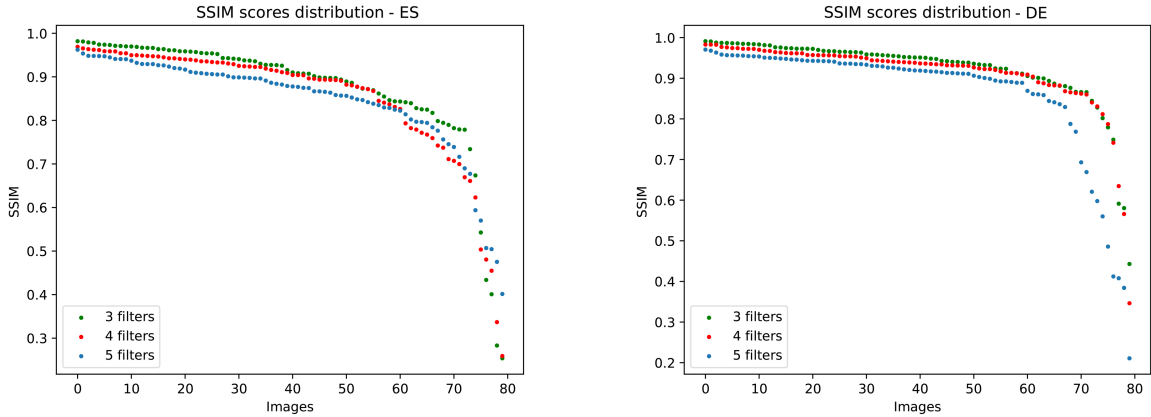


Figure 4.9: SSIM values distributions for the attacking images produced by ES (left) and DE (right) with 3, 4, and 5 filters. Adversarial images generated with DE have a higher SSIM score than the adversarial images generated with ES.

Optimization	3 filters	4 filters	5 filters
Inner-ES	18s	20s	22s
Inner-DE	65s	70s	75s

Table 4.8: Average time (seconds) for one outer generation for ES and DE.

The experimental results show that using a multi-objective strategy to craft adversarial attacks allows for the discovery of personalized image filters having different image properties and aesthetics and can be used to fool an emotion recognition model and thus can be used as a tool for privacy protection to filter out the emotional component for prying software. Moreover, by analysing the confusion matrices we observed that the behaviour of the model under attack aligns with the human perception of emotions. Studies on the perception of emotions [159–161] have shown that happiness is among the easiest emotion to recognize for humans, while emotions like contempt and neutral are more difficult to recognize. This opens up different research directions. For example, by leveraging the power of adversarial attacks, new tools could be developed to automatically analyse the perceptions of emotions without requiring human feedback which can be time consuming and expensive.

4.3 Universal Multi-objective Attack

In this section, we present a multi-objective variant of the proposed attack to generate universal unrestricted adversarial examples. Deployed deep-learning systems may be equipped with detection mechanisms to protect themselves from malicious activity. Thus, we propose to formulate the attack as a multi-objective optimization that takes into account not only the attack success rate but also the detection rate in order to craft undetectable adversarial images. Moreover, we are interested in finding a universal perturbation that, when applied to *any* image, can fool the classification model without any additional computation. An image-agnostic perturbation allows us to save computational time and effort since the optimization problem has to be solved only once for the whole dataset instead of each individual sample. Additionally, by forcing the attack to find a universal perturbation, we aim at increasing the generalization property. The experimental results show that a universal perturbation optimized on a small dataset is able to fool new unseen images with a high success rate. It is important to note that, in this work, we focus on universal adversarial perturbations crafted with respect to one dataset and one classifier. The extension to multiple datasets and classifiers will be considered for future work.

4.3.1 Problem formulation

Let x be an image, and let M be a neural network classifier that predicts the class label for the input image, s.t. $M(x) = y$.

In the case of *per-image* approaches, an adversarial attack attempts to find a different δ for each image that turns the image x in an adversarial image $x^* = x + \delta$ such that the classifier is misled into making a wrong prediction, i.e. $M(x^*) \neq M(x)$. In this case, it is necessary to run the optimization process for each image.

On the other hand, in the case of *universal* approaches the objective is to find *only one* such δ able to fool M for *almost all* the data points available in X , that is

$$M(x + \delta) \neq M(x), \quad \text{for almost all } x \in X \quad (4.12)$$

We model the attack as a multi-objective optimization problem which considers both the attack success rate as well as a detection mechanism bypassing rate. The goal is to give the

attacker the ability to bypass detection mechanisms. We believe this to be a powerful feature of our method given that the field of adversarial machine learning lacks such approaches.

Thus, we define multi-objective problem of our interest as

$$X^* = \operatorname{argmin}_{X'} \mathcal{F}_{MO}(X, X') \quad (4.13)$$

with

$$\mathcal{F}_{MO}(X, X') = \{1.0 - SR(X, X'), DR(X')\} \quad (4.14)$$

where

$$SR(X, X') = \frac{1}{|X|} \sum_{i=1}^{|X|} M(x_i) \neq M(x'_i), \quad (4.15)$$

is the attack *Success Rate* (SR) and

$$DR(X') = \frac{1}{|X|} \sum_{i=1}^{|X|} D(x'_i) \quad (4.16)$$

is the *Detection Rate* (DR), where D is the detection method that returns 1 if the image is detected to be an attack and 0 otherwise, x_i is the i -th image of the original dataset X and X' is the set of perturbed images x'_i obtained by applying the same sequence of filters to all the images in X . The goal is to find the sequence of filters that, when applied to X , generates X^* which best optimizes Equation 4.14.

4.3.2 Experimental Setup

Target model: We evaluate the proposed method by attacking the convolutional neural network proposed by Papernot et al. in [41] and used also in [23] to prove the effectiveness of their attack. The model is composed of a series of 2 convolutional layers having 64 3x3 filters paired with ReLU activation function and a max-pooling layer, 2 convolutional layers with 128 3x3 filters with ReLU followed by another max-pooling layer, 2 fully connected layers with ReLU and a softmax layer used for the final classification. This network was trained using the CIFAR-10 dataset which is a very popular benchmark image dataset consisting of 50000 training and 10000 testing color images with a resolution of 32x32, belonging to 10 different classes. Dropout was used in order to prevent overfitting, and momentum and

parameter decay were employed to guarantee model convergence.

Dataset: We used the CIFAR-10 testing set for training our algorithm and evaluating its effectiveness. The set was divided into two subsets: the first 200 images were used for the filter configuration optimization process and the remaining 9800 images were used for testing the adversarial attack. The optimization subset of images was chosen relatively small in order to measure the power of the universal attack.

Attack implementation details: The hyperparameters default values used to conduct the experiments were fixed as follows, where not differently specified: number of filters = 5, mutation probability = 0.5, population size = 10 for the outer algorithm. For the inner algorithms, we set the population size equal to 5 and the number of generations was fixed to 3, an initial learning rate of 0.1 and decay rate of 0.75.

Moreover, since the goal is to find a universal perturbation that can transform (almost) any image in an adversarial example, we need to optimize the filter combination with respect to all images in the dataset. However, running the optimization algorithm for each image is computationally expensive, thus we optimize the filters considering batches of images. Moreover, we run the algorithm over the dataset multiple times in order to ensure that the filters combination has been sufficiently optimized and that a good solution has been found. Therefore, for this experiment we set the number of epochs¹ = 3 and the batch size = 100, as preliminary results showed good performance with this setup (Figure 4.10).

We choose *Feature Squeezing* [38] as the detection method in the objective function used during the optimization process since it is one of the most popular and low-cost techniques that has been proven to achieve high detection rates (over 85% for CIFAR-10 and Imagenet dataset) against different famous state-of-the-art attacks, such as FGSM [17], DeepFool [25] and CW [23]. Specifically, to perform the detection we used the combination of features squeezers reported in [38] to work best for CIFAR-10 images: reduction to 5-bit depth, a local median smoothing and a non-local mean smoothing, and threshold to find the illegitimate

¹This is the equivalent of *generations*

images set to 1.7547 ².

4.3.3 Evaluation and Experimental Results

Selection of the training epochs, number of filters, and parameters range

Several experiments were carried out in order to estimate the best trade-off between the performance of the proposed method and computation time. We tested three inner optimization algorithms (GA, ES, and Rand-T defined in Section 3.3.2) with the above-listed parameters configuration except for the number of epochs which was set to 10. We analyzed their attack success rate (SR), feature squeezing detection rate (DR) and computation time. We observed that all inner optimizers had similar performance-time behavior. Moreover, we decided to stick to 3 epochs since it was producing good results while keeping the computational time fairly low. Figure 4.10 illustrates the attack and detection rate curve with respect to the number of epochs with ES inner optimizer.

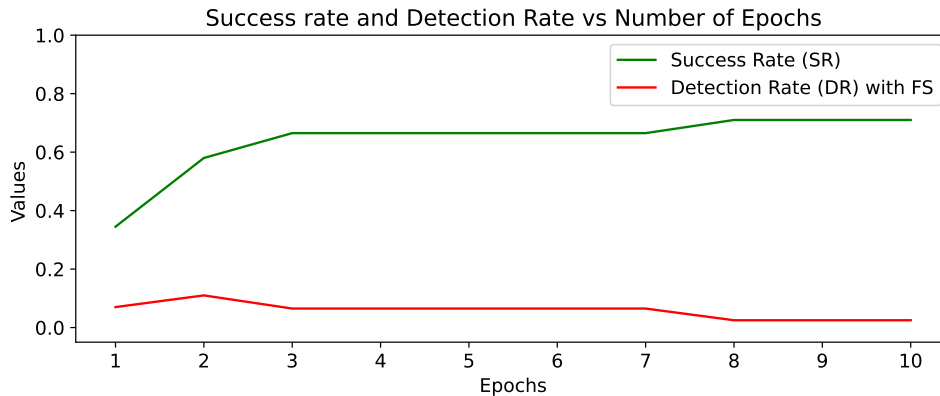


Figure 4.10: Success Rate (SR) and Detection Rate (DR) with Feature Squeezing w.r.t. Epochs with ES optimizer. The result show that 3 epochs are sufficient to obtain a good trade-off between attack success rate and detection rate.

Moreover, we also wanted to investigate the importance of choosing different numbers of filters for creating the adversarial configuration. The minimum filter selection was set to 3 while the maximum is the cardinality of set S of available filters. We adopted the

²https://github.com/mzweilin/EvadeML-Zoo/blob/master/Reproduce_FeatureSqueezing.md

policy of no-repeating filters, meaning that a filter can be picked only once inside a certain configuration in order to provide more diversity to the image manipulation. We calculated the attack rate of our algorithm by using all three inner optimization methods. Table 4.9 shows that using 5 filters has the best outcome in terms of attack success rate.

Table 4.9: Evaluation of attack success rate (SR %) with respect to the number of filters.

Optimizer	3 filters	4 filters	5 filters
ES	46.50	43.50	70.00
GA	58.50	52.00	68.50
Rand-T	41.50	45.50	61.00

In our implementation filters can be applied using different feature parameters similar to how Instagram allows users to control the effect of filters by manually adjusting their intensities within a certain range. The parameters of each filter can vary between a fixed range of values. The minimum and maximum values of each interval were found by performing a quality analysis on the modified images with the above-mentioned filters and diverse parameter values. This analysis allowed restricting the search space in order to further reduce the training time. In order to evaluate the universality of our attack we applied the optimized filter configuration to each image in the testing set and computed the detection rate defined as follows:

$$FSDR = \frac{1}{n} \sum_{i=1}^n D(x_i^*), \quad x_i^* \in \bar{X}^* \quad (4.17)$$

where D corresponds to the features squeezing detector which returns 1 if the image is identified as illegitimate and 0 otherwise, \bar{X}^* represents the set of successful adversarial examples, and $n = |\bar{X}^*|$.

In Table 4.10 we report the attack success rate and the detection rate for both training and testing subsets with the default hyperparameters values, which were found to work best.

First of all, from these results, we can note that, even if the attack success rate is lower than the ones obtained by other methods in literature (also greater than 90% in some cases), these values should be fairly compared to the ones obtained by the other methods excluding the attacks that would be blocked by a defense mechanism. For example, some of the most

Table 4.10: Attack success rate (SR %) and Feature Squeezing Detection Rate (FSDR %) with different optimizers on Carlini CNN and CIFAR-10 training and testing subsets, epochs = 3, number of filters = 5.



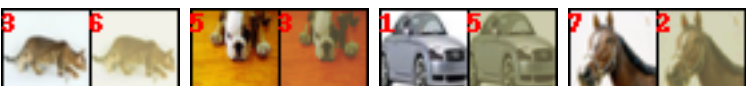




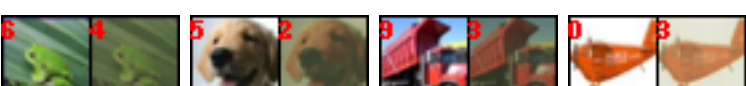
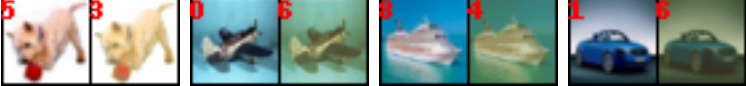



Optimizer	ASR % train set	FSDR % train set	ASR % test set	FSDR % test set
ES	70.0	2.1	63.7	3.5
GA	68.5	2.9	63.8	3.4
Rand-T	61.0	5.7	56.3	4.5

famous the state-of-the-art attacks like FGSM [17], BIM [19], DeepFool [25], CW [23] achieve 85%, 92%, 98% and 100% success rate ³, respectively on CIFAR-10 dataset. According to Xu et al. [38], Feature Squeezing reaches 20.8% detection rate on FGSM, 55.0% on BIM, 77.4% on DeepFool and 100.0% on CW. Although these are restricted white-box attacks, we report their success rate and detection rate values in order to evaluate the performance of our method under a broader view. Altogether Xu et al. [38] evaluated Feature Squeezing with respect to 11 different attacks on CIFAR-10 and reported an overall detection rate of 84.5% [38]. Considering this, our attack is very effective because among the successful adversarial images just very few attempts will be blocked by the defense mechanism. Moreover, we can observe a very good generalization ability of the model: when the adversarial perturbation generated by our method is applied to the test set (not used during the optimization process), we lose less than 10 percentage points for *SR*, maintaining also a very low detection rate when a defense mechanism based on Feature Squeezing method is applied.

Table 4.11 shows some successful adversarial examples generated by applying the filter configurations with their respective optimized parameters found by the proposed algorithm on the unseen images from the testing subset. For each adversarial example, we attached the original image and we also indicate the classification labels before and after the modification. It is very interesting to note that the solutions found by our method, i.e. the applied perturbations, are very uniform across the image and no unnatural patterns or high-frequency areas can be noticed.

³This results are reported from [38].

Table 4.11: Successful adversarial attacks on CIFAR-10 testing subset. On the left: original image; On the right: successful adversarial example.

Optimizer	Successful adversarial examples on the testing set
ES	
	
	
	
GA	
	
	
	
Tournament	
	
	
	
Label names	airplane : 0, automobile : 1, bird : 2, cat : 3, deer : 4, dog : 5, frog : 6, horse : 7, ship : 8, truck : 9

In summary, the experimental results show that the multi-objective method with detection feedback is able to produce successful adversarial examples while keeping the detection rate low. Even though the attack success rate is lower with respect to other state-of-the-art methods (restricted and unrestricted) we have the advantage of not being caught by detection methods. This indicates the potential of the proposed attack whose goal is not only to force the classifier to mispredict but also to evade possible defenses.

Chapter 5

Attacks on object detectors

In this chapter, we investigate the effectiveness of the proposed adversarial attack against state-of-the-art object detectors. We contextualize the problem of object detection in Section 5.1, and formulate the optimization problem in Section 5.2. We discuss the experimental results in Section 5.3.

5.1 Introduction

Object detection is the task of identifying and locating objects in an image or video, and has been successfully applied in a variety of real-world scenarios, such as autonomous driving [162], medical imaging [163], and intelligent surveillance in smart cities [164]. Such systems are highly security-sensitive, thus the robustness of these models is essential. While attacks to image classifiers have been extensively studied, the topic of attacks to object detectors remains largely unexplored, especially in the black-box scenario. The problem of generating realistic adversarial examples becomes more interesting and challenging since the number of targets that need to be attacked is much larger than in a pure classification task. In general, object detection models generate many bounding boxes for the same object. If one bounding box becomes unreliable, other bounding boxes may still work, making object detectors difficult to attack.

Another aspect that has not been adequately addressed for object detection is the use of adversarial machine learning techniques as defense methods against private information

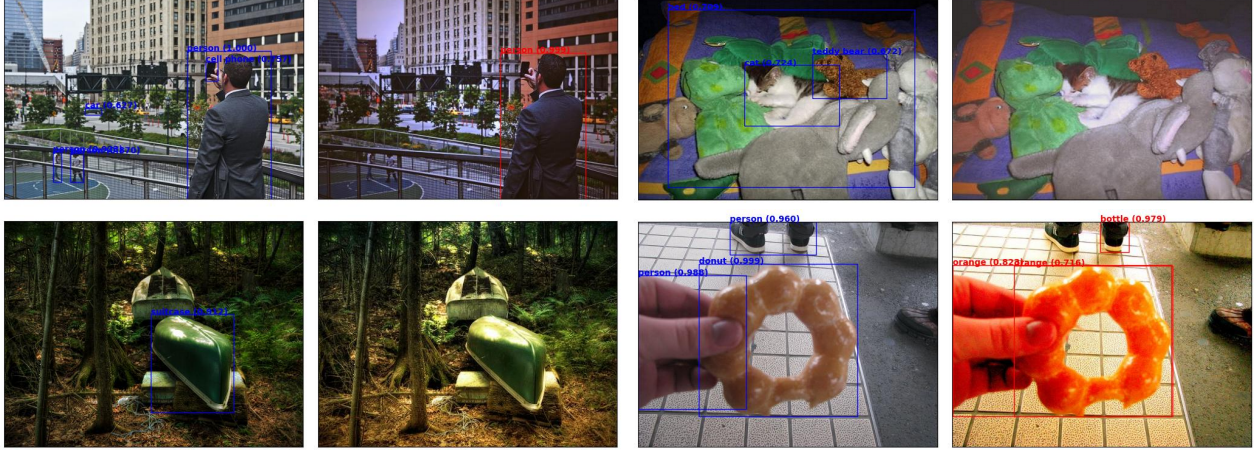


Figure 5.1: Adversarial images generated with our method on YOLOv3 (top row) and DETR (bottom row): clean images on 1st and 3rd columns, adversarial images on 2nd and 4th columns.

extraction, especially from images posted on social media. In this case, adversarial methods that produce natural-looking images become of primary importance in creating tools that block unauthorized information extraction and preserve privacy. It has been shown that information can also be easily extracted from datasets and learned models in the case of classification [42, 165–167].

Thus, we propose to attack object detectors with different architectures: YOLO (You Only Look Once) [106, 168] and DEtection TRransformer (DETR) [9]) to evaluate both the robustness of such models and the efficiency of our *One-for-Many* attack on a task more difficult than classification.

5.2 Problem Formulation

Formally, given an object detection model M , $x \in \mathbb{R}^{(H \times W \times 3)}$ a 3-channel RGB image with height H and width W and a list of predefined classes C , the object detection output is a list of K objects, each characterized by the pair (b_i, c_i) , where $b_i = ((x_{0i}, y_{0i}), (x_{1i}, y_{1i}))$ is a rectangular bounding box and $c_i \in C$ is a class label, as shown, for example, in Fig.5.1.

Given an image x , we formulate the adversarial problem on object detection as the problem to find x^* that minimizes the Object Detection Performance (ODP):

$$x^* = \operatorname{argmin}_{x'} ODP(x, x') \quad (5.1)$$

where

$$ODP(x, x') = \sum_{\substack{i=1 \\ Iou(b_i, b'_i) > 0.5 \\ c'_i = c_i}}^{|M(x)|} Iou(b_i, b'_i) \quad (5.2)$$

where $(b_i, c_i) \in M(x)$ are the bounding boxes and their corresponding labels returned by the object detection model M and $Iou(\cdot)$, the intersection over union measure, is defined as:

$$Iou(b_i, b'_i) = \frac{b_i \cap b'_i}{b_i \cup b'_i} \quad (5.3)$$

Moreover, since the aim of this study is to create natural-looking, artifacts-free adversarial samples, we state the problem as the following multi-objective optimization problem:

$$x^* = \operatorname{argmin}_{x'} \mathcal{F}_{MO}(x, x') \quad (5.4)$$

with

$$\mathcal{F}_{MO}(x, x') = \{ODP(x, x'), 1 - SSIM(x, x')\} \quad (5.5)$$

where $SSIM$ is the metric that quantifies image quality degradation of x' with respect to x and x^* represents the perturbed image that minimizes Equation 5.5.

5.3 Evaluation and Experimental Results

We evaluate the attack on the well known YOLO [106, 168] family of object detectors and DETR [9], a recently proposed detector with a transformer encoder-decoder architecture. We evaluate the effectiveness of the attack by analyzing its results with varying numbers of filters applied and assess the quality of the generated adversarial images. We also provide a comparison with other state-of-the-art methods.

Target models: The YOLO family of object detectors was chosen for our experiments. According to recent research, the YOLOv3 [106] is one of the most robust object detectors

against adversarial attacks [111]. Since it is one of the most commonly used models in the literature for object detection, this choice facilitates comparisons with other models. Furthermore, we investigate the robustness of YOLOv4 to check whether the robustness has improved over the previous version, and the robustness of the scaled models YOLOv3-tiny and YOLOv4-tiny [169] designed for low-end GPU devices. These last two models are built appositely to run with reduced resources and they are commonly used in embedded architectures where standard models cannot run. We used publicly available pre-trained networks provided in the official repository <https://github.com/pjreddie/darknet> and standard settings with input dimensions $608 \times 608 \times 3$ for YOLOv3, YOLOv4 and $416 \times 416 \times 3$ for YOLOv3-tiny, YOLOv4-tiny. Moreover, we use DETR [9] as a target model, because it is one of the best object detector models in terms of precision.

Dataset: We randomly selected 400 images from the MS COCO Val2017 dataset [107]. It is a large-scale dataset extensively used for training, testing, and evaluating the performance of object detection models. It contains 80 object categories, such as person, car, bird, and many more.

Hyperparameter configuration: We configured the optimization algorithm as follows: for the outer algorithm a population size of $N_{out} = 10$, generations $G_{out} = 3$ and a mutation probability $\rho = 0.5$; for the inner algorithm a population size of $N_{in} = 5$, the number of generations $G_{in} = 3$, initial learning rate = 0.1 and decay rate = 0.75. Experiments were run with 3, 4, and 5 filters.

Evaluation metrics: We use *precision*, *recall*, and *mean average precision (mAP)* metrics to evaluate the performance of the models before and after the adversarial attacks. Precision (p) and recall (r) are defined as:

$$p = \frac{TP}{TP + FP} \quad r = \frac{TP}{TP + FN} \quad (5.6)$$

where TP are the true positives, FP are the false positives, and FN are the false negatives.

A detection is considered a TP if the IoU is greater than a predefined threshold t and the

predicted class is correct. If either the IoU is less than t or the predicted class is incorrect, the detection is considered a FP. A ground truth bounding box that is not detected by the model is classified as a FN. In the COCO dataset, average precision AP_t is computed as the area under the precision-recall curve using a 101-point interpolation.

$$AP_t = \int_0^1 p(r)dr \quad (5.7)$$

Then, the mean Average Precision (mAP) is calculated by taking the mean AP over all the classes and over all the IoU thresholds:

$$mAP = \frac{1}{n} \sum_{k=1}^n AP_k \quad (5.8)$$

where n is the number of classes and AP_k is the mean AP of class k over all the IoU thresholds $t \in [0.5, 0.95]$ with a step size of 0.05. Common mAP variants are mAP_{50} and mAP_{75} where 50 and 75 represent the respective IoU thresholds.

Results: We assess the effectiveness of the attack by measuring the decrease in terms of mAP , mAP_{75} and mAP_{50} obtained by each model on the dataset modified with a different number of filters. In Figure 5.2 we report the distributions of mAP_{50} values for all tested models. Blue columns stand for values obtained testing the clean dataset, while orange columns stand for values obtained testing the dataset of images modified with 3, 4, and 5 filters. Comparisons with respect to the models can be made by reading the plots by columns, while comparisons with respect to the number of filters can be made by reading the plots by rows.

It is important to note that all the values shift towards lower values in the case of filtered images. In particular, for all combinations, the number of images with the highest mAP_{50} values significantly declines (blue columns in the right part of each plot are significantly higher than the orange ones), while the number of images with the mAP_{50} values significantly increases (orange columns in the left part of each plot are significantly higher than the blue ones). These differences get progressively more pronounced as the number of filters increases.

In Table 5.1 we show mAP , mAP_{50} , mAP_{75} , *precision* and *recall* for all models, before and after the attack. YOLOv3 confirms to be the most robust model, while both YOLOv3-

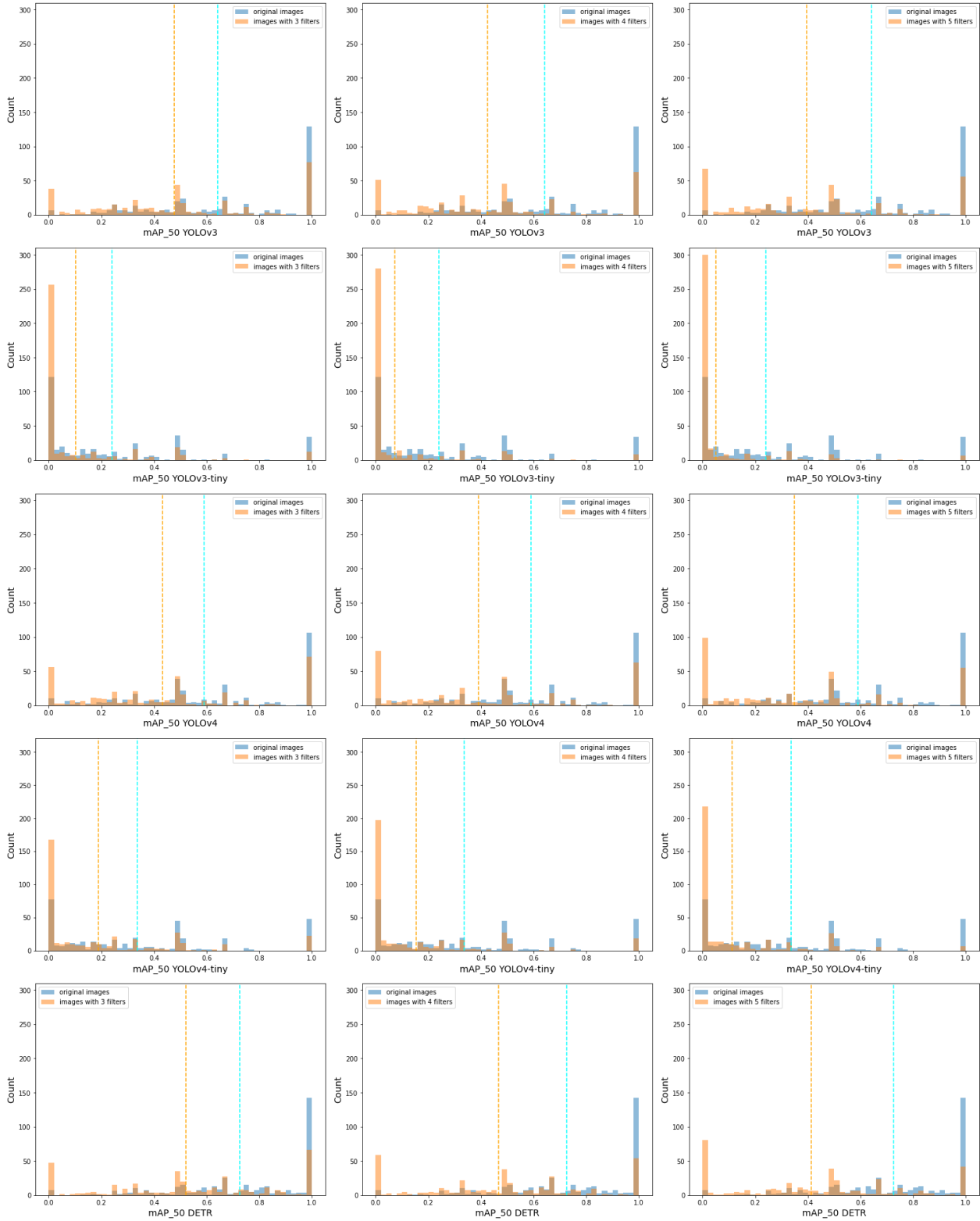


Figure 5.2: Distribution of AP₅₀ values obtained on YOLOv3, YOLOv3-tiny, YOLOv4, YOLOv4-tiny, DETR. The vertical blue dotted line represents the mean over all the mAP₅₀ values of the original images. The orange one represents the mean over all the mAP₅₀ values of the filtered images.

tiny and YOLOv4-tiny present an impressive decrease in robustness with respect to their original versions. The results on DETR are unexpected. Despite being the best performing on the clean dataset the mAP drops by 53% when using 5 filters. We observe similar behavior for precision and recall metrics.

Moreover in Table 5.1 we also provide the average values SSIM and NIMA to assess the quality of the adversarial images. The results show that overall the quality of the generated images is similar to the clean images.

The comparison with other state-of-the-art methods described in Section 2.2 should consider various factors. Some studies, such as [59, 111], utilize global mAP as a metric, while others, like [108], employ non-standard mAP metrics, making fair comparisons challenging.

In [59], the adversarial attack using Perling noise perturbations halves the mAP of the YOLOv3 model. This attack performs slightly better than ours but the patterns produced by procedural noise are evident, as shown in Figure 5.3. Lu et al. [111] reduces the performance of YOLOv3 to a $mAP = 0.24$ while considering only 20 categories of objects. Also in this case the adversarial patterns are highly noticeable. We achieve a mAP as low as 0.19 and a $mAP_{50} = 0.30$ on YOLOv3, outperforming also PRFA [110] which obtains a $mAP_{50} = 0.46$. The images generated with PRFA also exhibit heavy adversarial noise.

Moreover, none of the cited methods presented an evaluation of image quality, neither in terms of full-reference measures like SSIM nor in terms of no-reference measures like NIMA [64]. For example, in [59] the authors showed very good results in terms of mAP , but the images their algorithm produced do not have a good quality since they show visible patterns and artifacts (see Figure 5.3). This behavior is common to most of the restricted L_p -bounded attacks since L_p -norms are able to measure the absolute difference between the original image and the modified one, but they cannot capture in any way the image quality in terms of perception.

Finally, we compare the efficiency of the algorithms in terms of the number of queries. The methods proposed in literature need a large number of queries and, also in the case of systems built to work with limited access to the victim model, they require several thousands of queries to produce reliable attacks: $\simeq 30k$ for [108], $4k$ for PRFA [110]. Our algorithm requires a very low number of queries to find an attack: considering the query formula in

Table 5.1: Attack results for YOLO family and DETR. Precision and Recall values are calculated with IoU threshold of 0.5. For mAP, mAP_50, mAP_75, P_50, R_50 lower values indicate a stronger attack (\downarrow). For SSIM and NIMA higher scores indicate higher image quality (\uparrow).

Model	Filters	mAP	mAP_50	mAP_75	P_50	R_50	SSIM \uparrow	NIMA \uparrow
YOLOv3	clean	0.32	0.51	0.35	0.53	0.53	1.00	5.07
	3f	0.22 (-30%)	0.35	0.25	0.24	0.35	0.86	4.77
	4f	0.21 (-36%)	0.32	0.23	0.21	0.32	0.83	4.76
	5f	0.19 (-42%)	0.30	0.20	0.19	0.30	0.79	4.75
YOLOv3 tiny	clean	0.10	0.18	0.11	0.07	0.18	1.00	5.07
	3f	0.04 (-62%)	0.07	0.04	0.01	0.07	0.86	4.78
	4f	0.03 (-74%)	0.04	0.03	0.00	0.04	0.81	4.77
	5f	0.02 (-80%)	0.03	0.02	0.00	0.03	0.79	4.76
YOLOv4	clean	0.34	0.47	0.38	0.47	0.49	1.00	5.07
	3f	0.23 (-32%)	0.31	0.26	0.23	0.32	0.87	4.78
	4f	0.21 (-39%)	0.29	0.23	0.20	0.29	0.82	4.77
	5f	0.18 (-47%)	0.24	0.20	0.12	0.26	0.78	4.75
YOLOv4 tiny	clean	0.15	0.24	0.17	0.16	0.25	1.00	5.07
	3f	0.08 (-47%)	0.12	0.09	0.04	0.13	0.86	4.77
	4f	0.06 (-57%)	0.06	0.07	0.01	0.10	0.81	4.77
	5f	0.05 (-66%)	0.08	0.06	0.01	0.08	0.77	4.75
DETR	clean	0.44	0.63	0.46	0.72	0.70	1.00	5.07
	3f	0.29 (-33%)	0.41	0.30	0.38	0.46	0.85	4.77
	4f	0.26 (-40%)	0.38	0.27	0.32	0.42	0.82	4.76
	5f	0.21 (-53%)	0.31	0.21	0.24	0.36	0.78	4.74



Figure 5.3: Adversarial images from [59] (on the left) and [110](on the right).

Equation 3.7 and the parameters settings for the experiments, the maximum number of allowed queries is 490. The drawback of using an (apparently expensive) evolutionary approach is highly mitigated by the needed reduced number of generations and population size.

Transferability: Since it is usually unknown which architecture a detector uses, an attack is more effective if it is transferable among different detectors. Thus, for transferability, we evaluate the performance of the unseen detector using the adversarial images generated for the seen detector. In Table 5.2 and Figure 5.4 we show the results of attacking the YOLO family using the images generated with DETR. We obtain an average 27% drop in mAP across all models. We also notice an increase in the number of images with $mAP_{50} = 0$ after the attack. This suggests that the attack exhibits good transferability.

In conclusion, the experimental results show that our method can also be successfully employed to attack object detectors despite their high performance on clean images. Specifically, it outperforms state-of-the-art methods with minimum integration effort while also executing the attack in a more constrained setup, that is a black-box setting with low number of allowed queries. This is due to the fact that, despite their complex architecture, such models still rely on CNN-based components to extract the image features that are then used to detect the objects. Considering the vulnerability of CNNs, these object detectors also become prone to attacks. This emphasizes once again the high susceptibility of deep neural networks to image filtering and calls out for further investigation in order to propose more reliable models.

Table 5.2: Attack results when transferring adversarial images generated on DETR to YOLO family. Precision and Recall values are calculated with IoU threshold 0.5.

Model	Filters	mAP	mAP_50	mAP_75	P_50	R_50
YOLOv3	clean	0.32	0.51	0.35	0.53	0.53
	3f	0.26 (-19%)	0.41	0.28	0.39	0.43
	4f	0.25 (-22%)	0.40	0.27	0.35	0.41
	5f	0.22 (-31%)	0.36	0.25	0.30	0.30
YOLOv3 tiny	clean	0.10	0.18	0.11	0.07	0.18
	3f	0.08 (-27%)	0.13	0.07	0.03	0.13
	4f	0.07 (-32%)	0.12	0.08	0.03	0.12
	5f	0.06 (-39%)	0.11	0.07	0.03	0.11
YOLOv4	clean	0.34	0.47	0.38	0.47	0.49
	3f	0.29 (-13%)	0.41	0.32	0.39	0.43
	4f	0.28 (-18%)	0.38	0.31	0.30	0.40
	5f	0.24 (-27%)	0.34	0.28	0.25	0.36
YOLOv4 tiny	clean	0.15	0.24	0.17	0.16	0.25
	3f	0.11 (-23%)	0.19	0.13	0.08	0.19
	4f	0.10 (-32%)	0.16	0.11	0.05	0.16
	5f	0.09 (-42%)	0.14	0.09	0.04	0.14

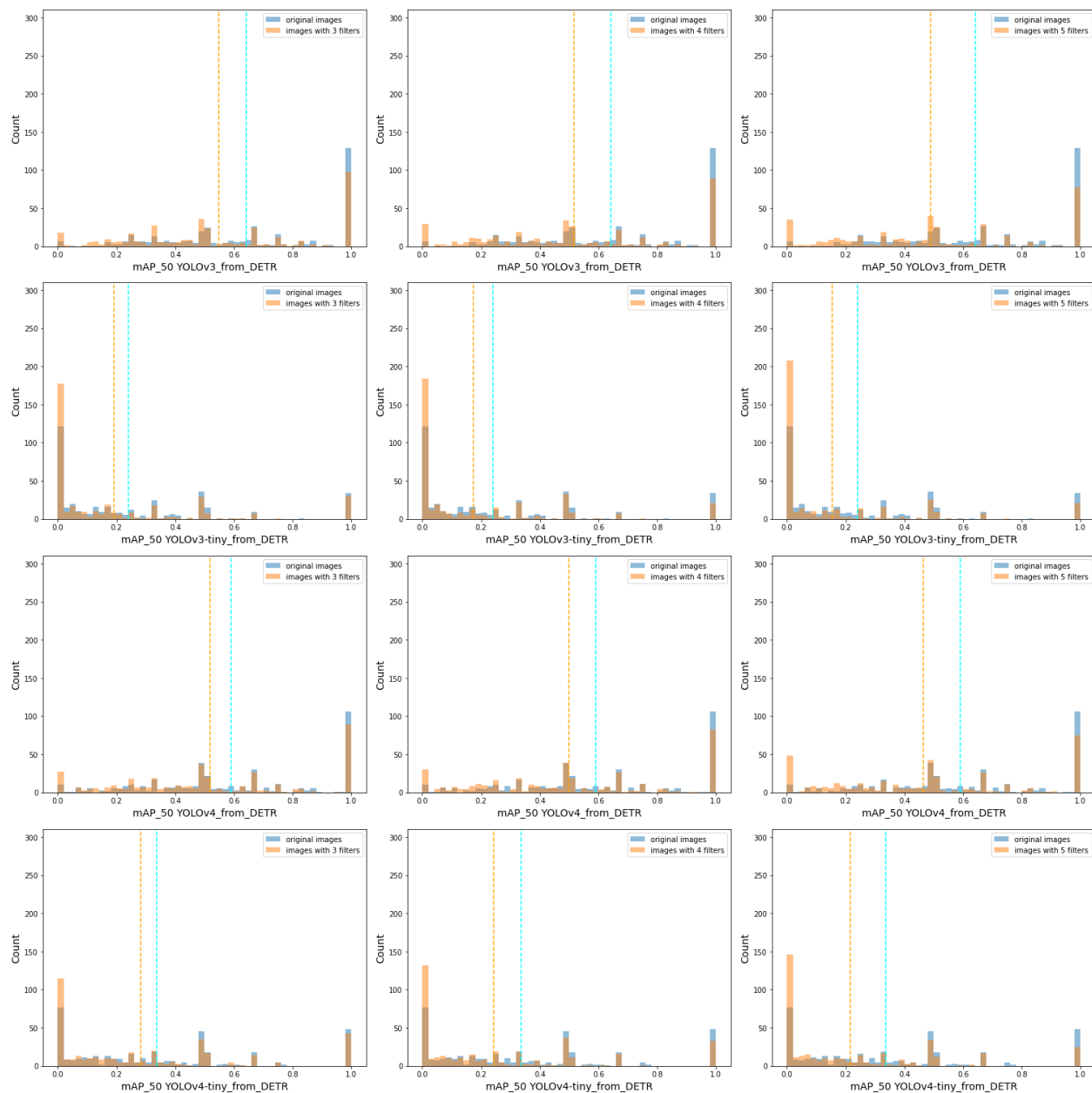


Figure 5.4: Results of transferability test: distribution of mAP₅₀ values obtained on YOLOv3, YOLOv3-tiny, YOLOv4, YOLOv4-tiny from DETR-optimized filters. The vertical blue dotted line represents the mean over all the mAP₅₀ values of the original images. The orange one represents the mean over all the mAP₅₀ values of the filtered images

Chapter 6

Attacks on multimodal explanations

In this chapter we assess the effectiveness of the proposed attack on a novel multimodal explanations model that aims to describe the underlying decision process of a neural network using natural language. We briefly introduce our approach in Section 6.1 and formulate the problem of attacks to XAI in Section 6.2. We present the experimental setup in Section 6.3 and discuss the results in Section 6.4.

6.1 Introduction

With this case study, we intend to evaluate the robustness of the textual explanations of the newly proposed transformer-based multimodal model NLX-GPT [126] for action recognition to black-box adversarial attacks. The NLX-GPT model takes an image in input and returns an output that contains an activity prediction, a textual explanation that justifies the prediction and a visual explanation map that highlights the most relevant image regions for the prediction. To the best of our knowledge this is the first work that studies the vulnerabilities of multimodal explanations systems against black-box unrestricted adversarial attacks. We generate the adversarial examples using the *One-for-Many* method. Moreover, to further reduce noticeability we propose to manipulate the images by operating differently on different image regions, e.g. avoiding perturbing sensitive areas such as the human skin. We also analyze the impact of the attack when focusing on regions most attended by the model. We show that naturally looking adversarial images can be used to manipulate the explanations

of a state-of-the-art model.

In particular, we study the robustness of the target system under two scenarios: (i) keeping the activity prediction the same and changing the (textual) explanations and (ii) changing the activity prediction and keeping the (textual) explanations similar. We do not consider the scenario of attacking both activity and explanations since this would be similar to attacking classifiers. According to [170] the prediction-explanation generation mechanism within an explanation system should be strongly associated: changing the activity classification implies a change in its explanation, if the prediction remains the same the explanations should not change. Therefore, our objective is to break this association by attacking one of the two mechanisms only, while keeping the other unchanged.

6.2 Problem formulation

Let x be an RGB image. Let M_E be an M-XAI model such that:

$$M_E(x) = \{s = (a, e), x_e\} \quad (6.1)$$

where s is a generated sentence that contains the activity prediction, $a = (a_1, a_2, \dots, a_p)$, and the textual explanation, $e = (e_1, e_2, \dots, e_n)$; a_i and e_j are words, and p and n are variable sentence lengths; and x_e is the visual component of the multimodal explanation.

We define an adversarial example for the explainable model, M_E , the image x^* , with $M_E(x^*) = \{s^* = (a^*, e^*), x_e^*\}$ considering two scenarios:

$$\text{Scenario 1: } a = a^* \wedge e \not\approx e^* \quad (6.2)$$

where the activities (decisions) are the same, but the explanations are different or

$$\text{Scenario 2: } a \neq a^* \wedge e \simeq e^* \quad (6.3)$$

where the activities (decisions) are different and the explanations are semantically similar.

We model the problem of finding x^* a multi-objective problem that accounts for two conflicting criteria: the quality of the textual explanation (Q_e) and the quality of the generated adversarial image (Q_x). Thus, we formulate the optimization problem as:

$$x^* = \underset{x'}{\operatorname{argmin}} \mathcal{F}_{MO}(x, x') \quad (6.4)$$

with

$$\mathcal{F}_{MO}(x, x') = \{Q_e(x, x'), Q_x(x, x')\} \quad (6.5)$$

where

$$Q_x = 1 - SSIM(x, x') \quad (6.6)$$

where x is the original image, x' is the corresponding modified image and $SSIM$ [63] is a full-reference image quality assessment used to control the applied perturbation. This objective remains the same for both attacking scenarios. We adapt the function of the explanation quality Q_e based on the attack scenarios.

Therefore, we define the objective on the textual explanation for the scenario $a \neq a^* \wedge e \simeq e^*$ as:

$$Q_e = 1 - \cos(E(e), E(e')) \mathbb{1}_{\{(a, a') : a \neq a'\}} \quad (6.7)$$

where $E(\cdot)$ is the vector embedding [171] of the textual explanation and

$$\cos(E(e), E(e')) = \left(\frac{\sum_{i=1}^n E(e)_i E(e')_i}{\sqrt{\sum_{i=1}^n E(e)_i^2} \sqrt{\sum_{i=1}^n E(e')_i^2}} + 1 \right) / 2 \quad (6.8)$$

with n being the size of the embedding vector, and $\mathbb{1}$ is the indicator function that returns 1 if the argument is true, else returns 0. The goal is to keep the explanations e and e' as similar as possible, by minimizing their difference while having different activity predictions. For the scenario $a = a^* \wedge e \simeq e^*$ we formulate Q_e as:

$$Q_e = 1 - [1 - \cos(E(e), E(e'))] \mathbb{1}_{\{(a, a') : a = a'\}} \quad (6.9)$$

with $E(\cdot)$ and $\cos(\cdot)$ as previously defined. In this case, the goal is to minimize the similarity between explanations while keeping the activities the same. We choose a cosine-based similarity measure with neural sentence embedding because it has been found to have the highest correlation with human judgment and to outperform other methods, such as METEOR or BLEU [122, 171, 172]

6.3 Experimental setup

Target model: We evaluate the attacks on the recent multimodal explanation model NLX-GPT for activity recognition [126], which textually explains its prediction using CLIP [173] as

vision encoder and the distilled GPT-2 pre-trained model [174] as decoder. NLX-GPT returns also a visual explanation (map) based on the cross-attention weights of the model. The distilled GPT-2 was pre-trained on image-caption pairs (COCO captions [107], Flickr30k [175], visual genome [176] and image-paragraph captioning [177]). NLX-GPT was fine-tuned on the activity recognition dataset ACT-X [127] (18k images). The visual encoder is fixed for both the pre-training and fine-tuning stages.

Dataset: We use the test set of the ACT-X [127], a 3,620-image dataset used to explain decisions of activity recognition models. Each image is labeled with an activity and three explanations. We perform the attack on the 1,829 images with correctly predicted activity by NLX-GPT.

Attack variants: We analyze four variants of our algorithm to consider different filter applications modalities and different objective functions: full image filtering with single (FL-s) and multi-objective (FL-m); localized image filtering with single (LC-s) and multi-objective (LC-m).

Full image filtering consists in applying the image filters on the entire image according to Equation 3.2. FL-s uses only the explanations quality (Q_e) while FL-m considers both explanations quality (Q_e) and image quality (Q_x) as defined in Equation 6.4.

We propose a localized filter application to further reduce the noticeability of the adversarial perturbation. We focus the perturbations in specific areas of the image I based on a partition defined by semantic segmentation into sensitive, $S = \bigcup R_i^s$, and non-sensitive regions, $\bar{S} = \bigcup R_j^n$, such that $I = S \cup \bar{S}$. Sensitive regions correspond to objects (e.g. human skin) whose unrealistic colors could raise suspicion, whereas non-sensitive regions can be modified more without making the image look unnatural.

First, we detect the semantic regions of an image using a state-of-the-art model [11]. Then, we identify skin areas ¹ to determine the sensitive areas S and mark them as unalterable. We follow up with a color-based oversegmentation [178] to further partition the semantic regions into smaller areas and obtain the non-sensitive regions \bar{S} .

¹Skin Segmentation Network

We binarize the visual explanation x_e to retain the most active parts, denoted as v_e . Next, we select only the relevant non-sensitive areas that contain highly active attention points, namely $\bar{S}_a = \bar{S} \wedge v_e$ for scenario $a \neq a^* \wedge e \simeq e^*$, while for scenario $a = a^* \wedge e \simeq e^*$ we modify the areas $\bar{S}_a = \bar{S} - v_e$ that do not contain highly active attention points. We denote the omitted areas as $\bar{S}_{na} = \bar{S} - \bar{S}_a$. To generate the perturbed image x' we apply to the clean image, x , a sequence of L filters:

$$x' = S \cup f_{k_1} \circ f_{k_2} \circ \dots \circ f_{k_L}(\bar{S}_a) \cup \bar{S}_{na}. \quad (6.10)$$

where each f_i is the parameterized version of a filter selected from a set $F = \{f_1, f_2, \dots, f_F\}$. The single objective localized attack (LC-s) considers only the explanations quality (Q_e), while the multi-objective uses also the image quality (Q_x) as defined in Equation 6.6.

Method for comparison: As method for comparison we adapt ColorFool [31] to our problem. ColorFool defines four types of sensitive regions: person, sky, vegetation, and water using a cascade segmentation module [31]. Adversarial images are generated by modifying the colors of the regions in the perceptually uniform *Lab* color space within specific ranges defined based on image semantics and human perception. Specifically, we extend ColorFool to consider also the explanation quality Q_e , defined as:

$$Q_e = \cos(E(e), E(e')) \quad (6.11)$$

with $\cos(\cdot)$ and $E(\cdot)$ previously defined in Equation 6.8. For scenario $a \neq a^* \wedge e \simeq e^*$ we return the perturbed image x' that maximizes Q_e , while for scenario $a = a^* \wedge e \simeq e^*$ we return x' that minimizes Q_e , computed over a predefined number of trials. We refer to this method as ColorFoolX (CFX).

Attack implementation details: We compare different filter application approaches and objective functions for a fair comparison with ColorFoolX, which does not directly account for image quality during the attack. For ColorFoolX we allow a maximum of 1000 trials. For the multi-objective evolutionary method, the size of the outer population is $N_{out} = 10$, the number of outer generations is $G_{out} = 10$, and the mutation probability is $\rho = 0.5$. For the



Figure 6.1: Adversarial images generated for a clean image (top left) for scenario $a \neq a^* \wedge e \simeq e^*$ by LC-m: localized filtering with multi-objective (second column) and CFX: ColorFoolX (third column) and the visual attention maps corresponding to the activity prediction (second row). The images have different activity - clean image: *manual labor*, LC-m: *driving*, CFX: *driving tractor*. The textual explanation is the same for all images: *he is bent over and pushing a tire with his hands*. The MANIQA scores are 0.709, 0.701, 0.705 for clean, LC-n, and CFX, respectively.

inner optimization, we use ES with a population size $\lambda = 5$, generations $G_{in} = 3$ with initial learning rate $lr = 0.1$ and decay rate $\beta = 0.75$.

6.4 Evaluation and Experimental results

We assess the quality of the explanations generated for the adversarial images using cosine similarity to assess the (dis)similarity between explanations of successful adversarial images and their corresponding clean image. We evaluate the quality of the adversarial images with MANIQA [66]. We also analyze the colorfulness [179] of the adversarial images and compare it with the colorfulness of original images.

The colorfulness metric aims to reflect color vividness in accordance with human perception. Given an RGB image, the colorfulness is computed as follows:

$$C = \sigma_{rgyb} + 0.3 \cdot \mu_{rgyb} \quad (6.12)$$



Figure 6.2: Adversarial images generated for a clean image (top left) for scenario $a = a^* \wedge e \approx e^*$ by LC-m: localized filtering with multi-objective (second column) and CFX: ColorFoolX (third column) and the visual attention maps corresponding to the activity prediction (second row). The images have the same activity *ballroom* but different explanations - clean: *she is standing on a dance floor dancing with a partner*; LC-m: *he is standing on a stage with a ball in his hands*; CFX: *she is standing on a stage with a group of people*. The MANIQA scores are 0.689, 0.662, 0.690 for clean, LC-m, and CFX, respectively.

where

$$\sigma_{rgyb} = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2} \quad (6.13)$$

$$\mu_{rgyb} = \sqrt{\mu_{rg}^2 + \mu_{yb}^2} \quad (6.14)$$

where σ and μ are the standard deviation and the mean value of the pixels along direction (\cdot) , and

$$rg = R - G \quad (6.15)$$

$$yb = \frac{1}{2}(R + G) - B \quad (6.16)$$

where R, G, B are the red, blue, and green channels.

We measure the success rate (S_r) of an adversarial attack as:

$$S_r = \frac{1}{N_a} \sum_{j=1}^{N_a} \mathbb{1}_\omega \quad (6.17)$$

where N_a is the total number of images and, for $a \neq a^* \wedge e \approx e^*$:

$$\omega \triangleq \{(a_j, a_j^*) : a_j \neq a_j^* \wedge \cos(E(e_j), E(e_j^*)) \geq t\} \quad (6.18)$$

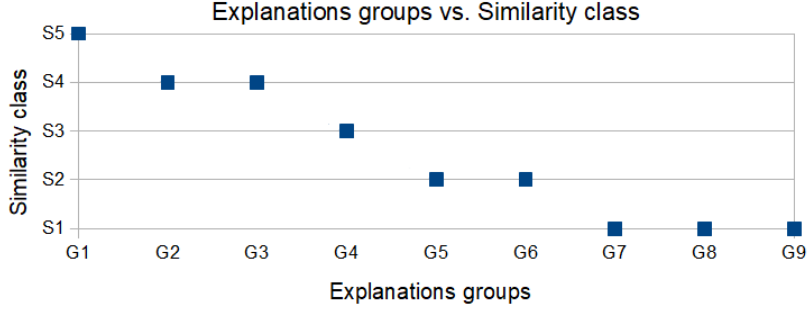


Figure 6.3: Mapping between explanation groups and similarity classes. KEY – Similarity classes- S1: not similar at all, S2: a little similar, S3: somehow similar, S4: very similar, S5: they are the same. Explanations pairs with $\cos(E(e), E(e^*)) > 0.85$ (i.e. G1-G3) are rated as highly similar.

where t is a threshold; and, for $a = a^* \wedge e \approx e^*$:

$$\omega \triangleq \{(a_j, a_j^*) : a_j = a_j^* \wedge \cos(E(e_j), E(e_j^*)) < t\}. \quad (6.19)$$

We determined the value of t with a subjective human evaluation of the similarity of explanation pairs. We created 9 groups for the explanations based on their cosine similarity, such that $G_i = \{(e, e^*) : \cos(E(e), E(e^*)) \in (1 - 0.05i, 1 - 0.05(i - 1))\}$ with $i \in \{1, 2, \dots, 8\}$ and $G_9 = \{(e, e^*), \cos(E(e), E(e^*)) \in (0, 0.6]\}$. From each group, we randomly selected 10 (e, e^*) pairs that were rated on semantic similarity on a 5-level Likert scale: *not similar at all*; *a little similar*; *somehow similar*; *very similar*; and *they are the same*. We used majority voting to assign each pair of explanations to a similarity class. Likewise, we labeled each group with the most frequent similarity class of the questions within the group. Eleven people who did not see the data prior to the test rated the similarity and could change their rating before completing the test. The mapping between explanation groups and similarity classes is shown in Fig. 6.3. We observe a decrease in semantic similarity starting from group G4, which corresponds to $\cos(E(e), E(e^*)) < 0.85$. Thus, we set the threshold $t = 0.85$.

Table 6.1 reports the success rates for all methods under both scenarios. Methods considering only the explanation quality (i.e. CFX, FL-s) achieve the best results in terms of success rate with an S_r of 64.62% for CFX and 63.09% for FL-s for the $a \neq a^* \wedge e \simeq e^*$ and an S_r of 73.82 for CFX and 77.53 for FL-s in the scenario $a = a^* \wedge e \approx e^*$. These methods apply the perturbation across wider areas of the image which allows heavier modifications. As we focus

on more localized areas (i.e. LC-s) and limit the freedom of the attack the S_r decreases. This behavior could also be caused by the noisiness and inaccuracy of the cross-attention-based visual maps (x_e) which may fail to accurately point to the areas relevant for the prediction. Since we use the visual maps to localize the areas to attack, if the visual maps are imprecise it leads to the selection of areas that are not as relevant for the prediction. Thus, it results in a lower attack success rate. Therefore, these model-intrinsic attention maps require more investigation to fully assess their relevance for localized adversarial attacks.

Table 6.1: Success Rate (S_r %; the higher, the better) for the scenarios $a \neq a^* \wedge e \simeq e^*$ and $a = a^* \wedge e \approx e^*$. KEY – CFX: ColorFoolX, LC-s: localized filtering with single objective, FL-s: full image filtering with single objective, LC: localized filtering with multi-objective, FL: full image filtering with multi-objective.

Scenario	CFX	LC-s	FL-s	LC-m	FL-m
$a \neq a^* \wedge e \simeq e^*$	64.62	51.33	63.09	43.47	47.62
$a = a^* \wedge e \approx e^*$	73.82	67.47	77.53	51.76	49.45

We further notice a decrease in attack performance as we increase the constraints enforced on the optimization process. In this case, on top of the localized-based restriction, we also limit the amount of the perturbation applied using SSIM. In this case, the algorithm has to find an optimal trade-off between explanation quality and image quality which is not always feasible given the competing nature of the two objectives. We also notice that the methods are more effective in the scenario $a = a^* \wedge e \approx e^*$, achieving a S_r of up to 77.53% for FL-s. In this case, the selected alterable areas are more numerous since we focus on regions that are not highly attended by the explanation model, and thus in general the adversarial perturbation is more spread across the image. Overall, the results show that both our method and CFX are able to change the model’s behavior and break the correlation between activity prediction and its explanations in two different scenarios. If the intent of the adversary is to only undermine the reliability of the explanation, then scenario $a = a^* \wedge e \approx e^*$ should be used since it has a higher attack success rate. If the attacker’s objective is to also change the activity prediction, scenario $a \neq a^* \wedge e \simeq e^*$ should be considered.

Although both CFX and evolutionary-based methods (LC and FL) produce comparable

Table 6.2: MANIQA scores of adversarial examples and their corresponding clean version. LC-m, FL-m, and CFX generated images with the highest quality score.

Attack \ Scenario	$a \neq a^* \wedge e \simeq e^*$		Clean	Adversarial
	Clean	Adversarial		
CFX	0.70	0.68	0.70	0.67
LC-s	0.69	0.66	0.70	0.65
FL-s	0.70	0.65	0.70	0.65
LC-m	0.69	0.67	0.70	0.68
FL-m	0.70	0.66	0.70	0.68

results within similar experimental setups, the generated adversarial images have different visual characteristics and aesthetics. In general, the evolutionary-based attack produces images with more toned-down soft vintage looks while most of the images generated by ColorFoolX have vivid colors (see for example are shown in Fig. 6.1 and Fig. 6.2). We quantitatively evaluate the perceptual quality of the adversarial images with MANIQA $\in [0, 1]$ (the higher, the better). The average MANIQA scores (Table 6.2) varies from 0.65 for FL-s and LC-s to 0.68 for CFX, LC-m, and FL-m while the average score on the clean images is 0.70. This indicates that the adversarial perturbations do not degrade the quality as perceived by MANIQA. Moreover, we observe that in the case of localized evolutionary attack (LC) the visual attention maps relative to the activity prediction are less noisy while in the case of CFX the attention is distributed over the whole image. It appears that the localized attack forces the model to focus on specific areas instead of having its attention scattered all over the image.

In Figures 6.5, 6.6 we show the distribution of colorfulness scores of adversarial images and their corresponding original version. The higher the scores, the more colorful the image is. We observe that LC-m and FL-m generate images with colors most similar to the original images, whereas LC-s and FL-s tend to generate images with more washed out muted-colors. This indicates that the image quality objective contributes towards the generations of more

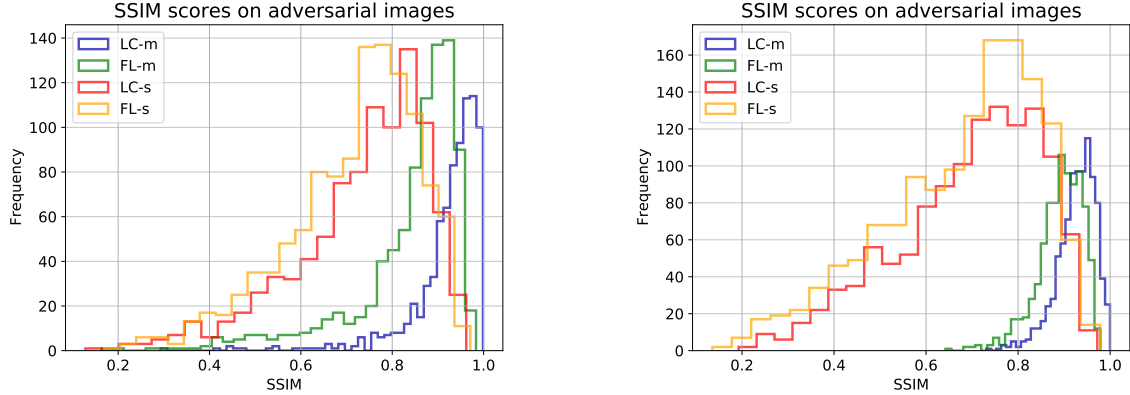


Figure 6.4: SSIM of adversarial images for scenario $a \neq a^* \wedge e \simeq e^*$ (left) and for scenario $a = a^* \wedge e \simeq e^*$ (right). The multi-objective attacks generate images with the highest SSIM.

natural² looking images, as also shown by the SSIM scores in Figure 6.4. On the contrary CFX, as its name suggests, generates very colorful images that diverge the most from the natural distribution. However, images different from the original ones does not necessarily imply worse quality since automatic image quality assessment metrics are not always reliable. Sometimes they may favour certain image characteristics over others and do not align with human perception. Thus, a human subjective evaluation remains the best measure to assess perceptual realism, which we will address in future studies.

In order to choose a text similarity metric for the proposed attack, we considered different automatic evaluation metrics for natural language generation tasks such as METEOR and cosine-based measures. In the following, we report the observations regarding METEOR.

We evaluate the similarity between pairs of explanations (e, e^*) using METEOR [180]. We found that METEOR, despite using stemming and synonym matching, cannot capture the semantic similarity. This is mostly due to the alignment penalty which penalizes sentences that have correct words but in a different order. For example, given explanation $e = he is standing on a bridge with a backpack on his back$ and $e^* = he is wearing a backpack and standing on a bridge$, the METEOR score is 0.45 even though the two explanations have the same meaning. However, the cosine similarity is 0.97 which indicates that cosine-based evaluation correlates much better with human judgment, which was confirmed by different studies [122, 171, 172] and also by our subjective study. This behavior is shown in Figure 6.7

²with natural we intend images that resemble the original, non-manipulated images.

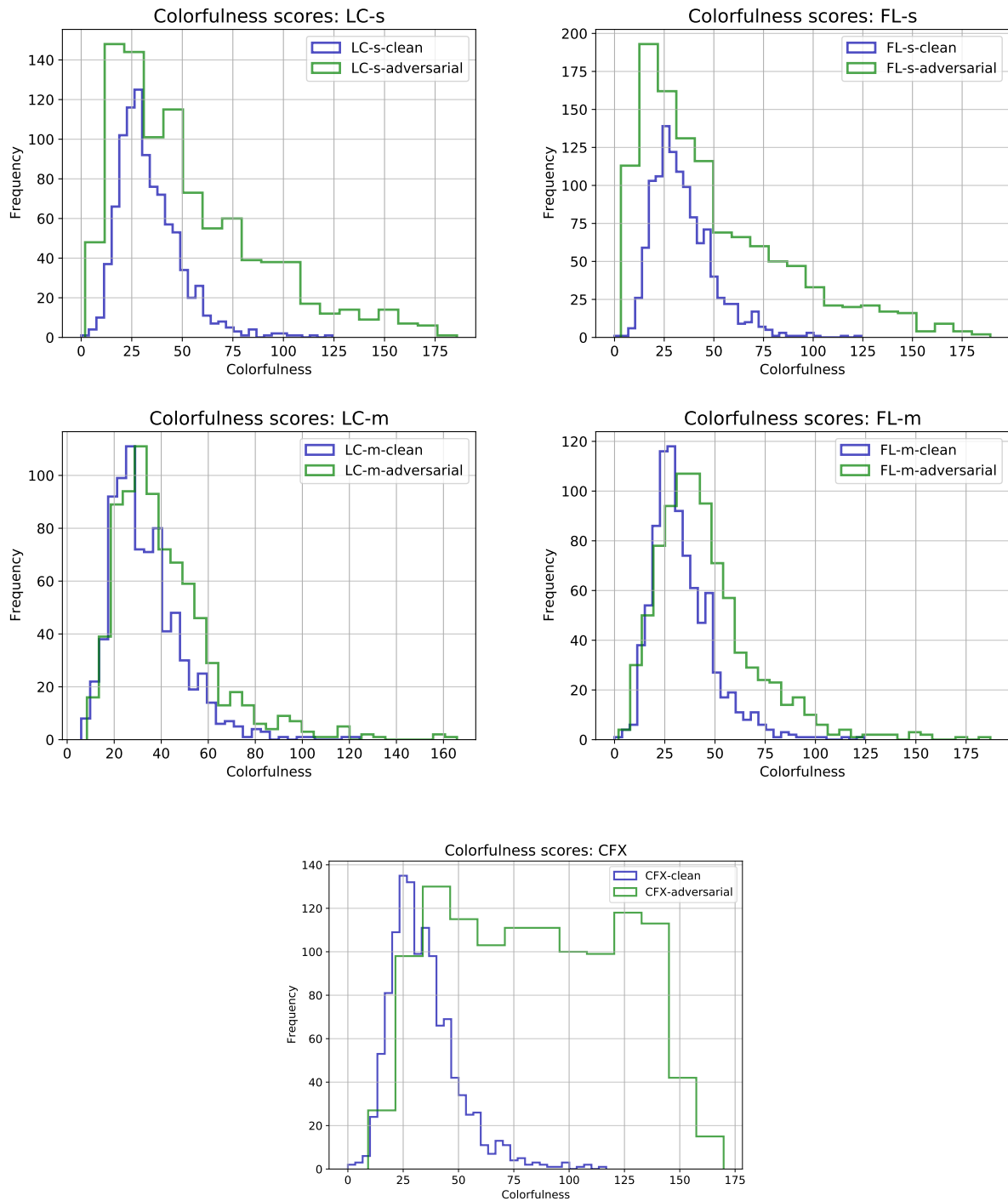


Figure 6.5: Colorfulness scores distribution for scenario $a \neq a^* \wedge e \simeq e^*$. LC-m and FL-m produce adversarial examples with colors most similar to the original images. CFX generates adversarial examples with colors quite different from the original distribution. The higher the score, the more colorful the image.

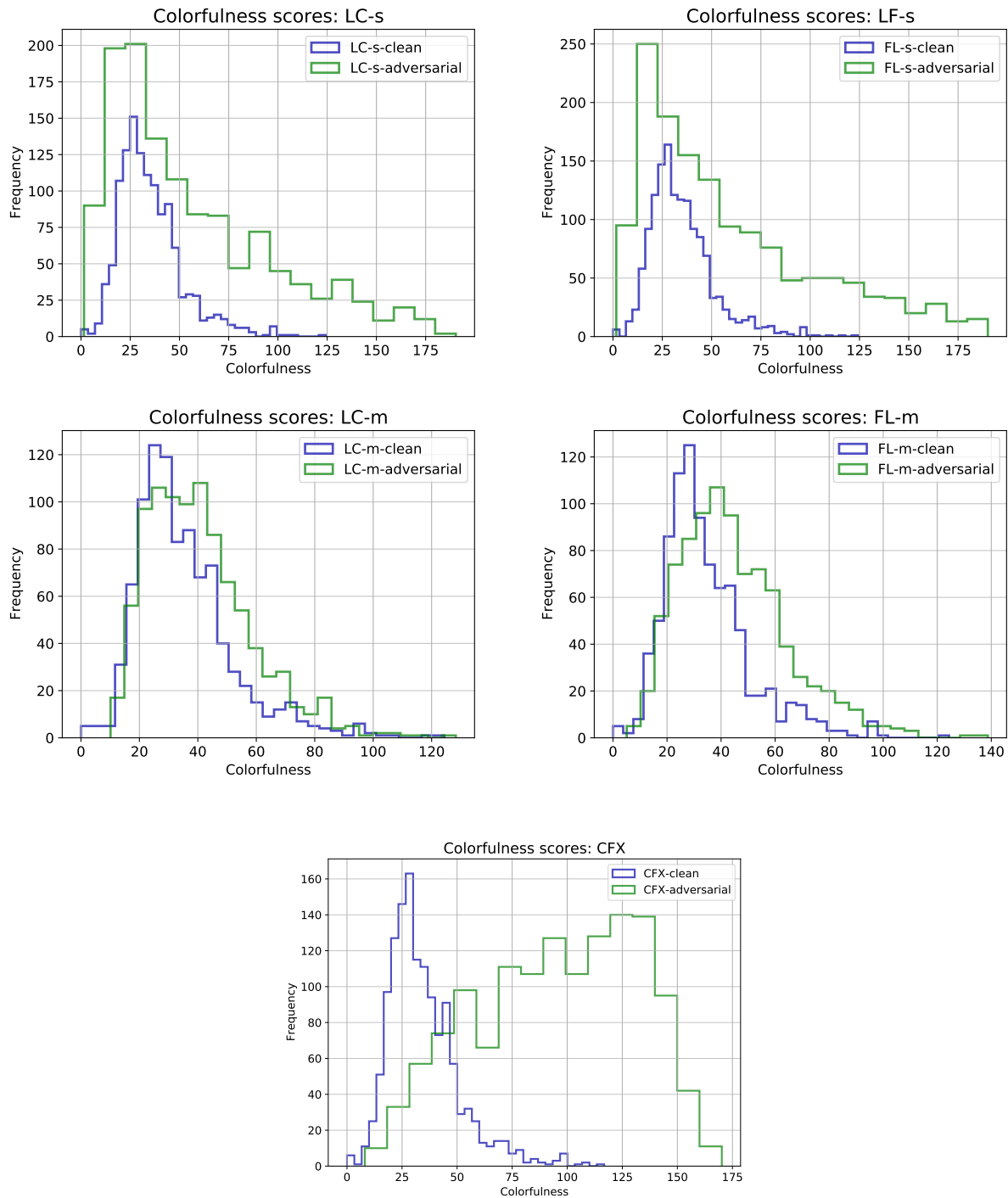


Figure 6.6: Colorfulness scores distribution for scenario $a = a^* \wedge e \approx e^*$. The adversarial examples generated with LC-m and FL-m have the colors most similar to the original images. In the case of CFX, the colors of adversarial examples diverge the most from the original distribution. The higher the score, the more colorful the image.

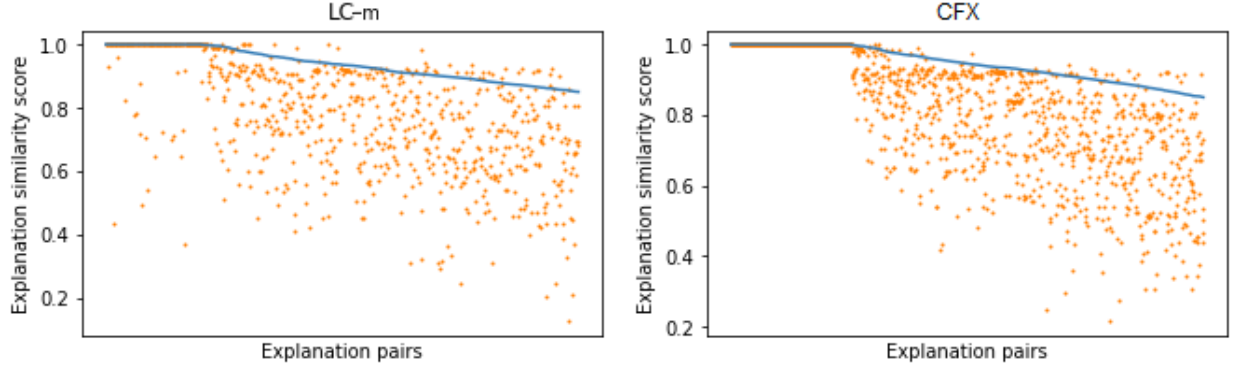


Figure 6.7: Explanation similarity scores computed with cosine similarity (■) and METEOR (■), generated with LC-m and CFX attacks for scenario $a \neq a^* \wedge e \simeq e^*$.

where explanations with high cosine similarity have low METEOR scores, showing that this metric is not suitable for such a task.

We also verify the contribution of each objective function in terms of success rate and SSIM scores for scenario $a \neq a^* \wedge e \simeq e^*$.

We start with a random approach, where we randomly perturb the images while only considering changing the activity prediction, disregarding explanation similarity and image quality. Then we consider each objective separately. For the image quality objective (Q_x) the aim is to find the image that changes the activity prediction with the highest SSIM. For the explanation objective (Q_e), the goal is to find an image that changes the activity prediction and has the highest explanation similarity. This corresponds to the single-objective attack previously discussed in Section 6.3. When using both objectives, the goal is to find an adversarial image that changes the activity predictions and has high explanation similarity and high image quality. This is equivalent to the multi-objective formulation presented in Section 6.2. We consider both full image filtering (FL) and localized image filtering (LC). Table 6.3 shows the S_r (Equation 6.17) and the SSIM between the adversarial images and their original version. In the case of the image quality objective, the adversarial images have the highest SSIM scores. However, the S_r is low. On the other hand, the textual explanation objective achieves the highest S_r at the expense of the image quality. This is the main justification for using the version with both objectives to find a trade-off between S_r and SSIM.

Text Quality	Image Quality	LC		FL	
		S_r	SSIM	S_r	SSIM
✓	✓	43.47	0.92	47.62	0.84
✓		51.33	0.73	63.09	0.72
	✓	26.93	0.95	28.16	0.84
		21.87	0.86	22.86	0.73

Table 6.3: Success Rate ($S_r\%$) and SSIM obtain when using different objective functions for scenario $a \neq a^* \wedge e \simeq e^*$.

We conducted a similar analysis for ColorFoolX to assess its multimodal attack capabilities with respect to the original version ColorFool. ColorFoolX searches for the adversarial examples that satisfy two conditions: one regarding the activity prediction and the other one related to the explanation similarity. For scenario $a \neq a^* \wedge e \simeq e^*$, CFX searches for an image with a different activity class and highest explanation similarity, while ColorFool would consider only the activity prediction. As a matter of fact, when considering only the activity prediction, the S_r is $\approx 80\%$. When considering also the explanation similarity (Equation 6.17) CF reaches $S_r = 37.23\%$, whereas CFX reaches $S_r = 64.62\%$. Thus, the CFX version is more beneficial for the multimodal attack.

We also briefly examined the correlation between activity prediction and explanation. We notice that when only attacking the activity the explanations tend to change. For example, using a full filter application with our method and focusing on changing only the activity, $\approx 66\%$ of images with different activity have an explanation similarity < 0.85 . However, this requires more investigation since currently there are no well-established procedures and evaluation metrics to assess the degree of a model’s faithfulness and opens up an interesting research direction for the future.

Conclusions

Summary of achievements

Deep neural networks (DNNs) have witnessed a significant progress over the last decade and have been successfully applied to a variety of applications in different domains. Despite their impressive performance, DNNs are vulnerable to adversarial attacks.

In this work, we have proposed a black-box adversarial method that generates unrestricted adversarial perturbations using Instagram-inspired image filters. The attack uses an evolutionary-based optimization to find the optimal perturbation that misleads a neural network. The method was designed to tackle some of the limitations of current state-of-the-art adversarial attacks: limited robustness of restricted attacks to defense mechanisms, unnatural perturbations produced by unrestricted attacks or the necessity of using additional resources to reduce noticeability. Moreover, it can be easily adjusted to many computer vision problems and allows for different types of attacks.

We validate our method on three computer vision tasks: image classification, object detection, and multimodal explanation for activity recognition. In the case of per-instance single objective attack on classification, we compare our method with ColorFool [31], a state-of-the-art unrestricted black-box attack based on color modifications. Our attack reaches up to 95.9% success rate, comparable results with ColorFool. However, our method significantly outperforms ColorFool on the transferability aspect, achieving an average transferability rate of 46.11% versus 29.05% obtained by ColorFool. Moreover, our attack is more robust to defiltering defense techniques. When considering the deception capabilities in the evaluation of the success rate, the proposed attack reaches a $SR@5$ up to 40% whereas ColorFool becomes completely ineffective, with only 1.2%.

For image classification task we also analyzed different attack configurations. We tested a multi-objective approach and included an image quality metric to control the amount of the applied perturbations when working with data highly sensitive to manipulations, such as facial images, and where excessive modification cannot bypass a human observer and easily raise suspicion. Our method effectively conceals the true emotions depicted in the images and could be employed as a tool for privacy protection. Furthermore, we used our algorithm to find universal perturbation that can bypass detection methods. The results obtained were promising but further investigations are needed in order to fully assess the attack’s ability under this configuration. Moreover, the optimization process could benefit more from using detection methods specifically designed for unrestricted attacks. We will consider this for future work.

We also evaluated the performance of our filter-based attack on object detectors, whose vulnerabilities still remain largely unexplored, especially in black-box setups. For the experimental phase we considered the well-known YOLO family of detectors and DETR, a recent model based on transformers. The performance drop varies from 42% on Yolo-v3 to 80% on Yolo-v3-tiny while the performance of DETR decreased by 53%. Surprisingly, DETR is less robust than Yolo-v3 despite being newer and more innovative. Our attack exhibits good transferability also in the case of object detectors, where adversarial perturbations have been found to generalize well across models with different architectures.

The final case study involves assessing the effectiveness of our method on a newly proposed multimodal explanation model. The model takes an image in input and simultaneously predicts an activity class and generates a textual explanation and a visual explanation map to show the reasoning process that led to that prediction. We empirically demonstrate that our method produces natural-looking images capable of breaking the correlation between activity and explanation under two different scenarios: keeping the activity the same while attacking the explanation and changing the activity while keeping the explanation semantically similar. We formulated the optimization problem considering different objective functions to work with different data types (i.e. image data and text data). Based on the adversary’s intent, the attack can be guided to focus more on disrupting the activity-explanation correlation or to also consider the quality of the adversarial images and find a trade-off between

the two. The success rate varies between 43.57% and 77.53%. Our method falls behind in performance compared to CFX, a variant of ColorFool [31] adapted for this task, when the adversarial perturbation is applied locally to specific regions in the image. This difference can be attributed to the over-segmentation process employed by our method which identifies smaller non-sensitive regions in contrast to CFX, and consequently less adversarial perturbation is applied. However, in the case of full image filtering our results align with those achieved by CFX. These are very interesting results that motivate us to further explore this topic.

Future work

Different directions could be considered for future work, focusing on both algorithmic advancements and potential applications. From the methodology perspective, a more comprehensive analysis of the filters and their adversarial behaviour could enhance the method’s performance. For example, conducting an in-depth examination of the frequency and order of each filter in the adversarial combinations could lead to the discovery of valuable information, such as identifying filters that are essential for generating effective adversarial perturbations. These findings could be integrated as prior knowledge into the algorithm to further reduce the number of queries required to execute an attack. Moreover, the power of multi-objective optimization could be further exploited to craft targeted attacks or to boost the deceitfulness of adversarial perturbations.

Additionally, we aim to extend the pool of attacked tasks, for instance including image segmentation models. By addressing the challenges introduced by the complexity of such models we can contribute to deepen the understanding of adversarial robustness and the potential implications of adversarial attacks in various domains.

Another exciting area for exploration is the generation of adversarial image filters with diffusion models that have recently demonstrated tremendous success in many fields such as image, video and speech generation, robotics, 3D modeling and neuroscience research. By taking advantage of the power of diffusion models we could craft diverse and visually appealing adversarial filters. An extension of this work could consist of integrating large

language models for text understanding to condition the generation of filters on user-provided text. This approach would allow users to customize the filters by specifying the characteristics and the desired visual effect according to their needs and preferences, opening up a new level of flexibility and control.

Given the recent emergence of the topic, the study of attacks on multimodal explanation models has uncovered several areas of interest for further investigation. For example, we could analyze the semantic similarities between activities and incorporate an additional objective function to explicitly measure and increase the dissimilarity between activities. This could provide some additional insight about the degree of correlation between activities and explanations. We also want to explore how the information provided by the visual explanation map can be used directly into the optimization process.

Finally, the study has revealed the lack of methods to evaluate the faithfulness of explanations. Thus, another possible research line could be designing model-agnostic evaluation methods to allow for comparative analysis of different explanations models.

Appendix A

Image filters

We show the effects of image filters in Table A.1 and Table A.2.

Table A.1: Effects of filters with different intensity β values and $\alpha = 1$

Filter	Original	$\beta = 0.5$	$\beta = 0.75$	$\beta = 1.0$	$\beta = 1.25$	$\beta = 1.5$
Clarendon						
Gingham						
Juno						
Reyes						
Lark						
Hudson						
Slumber						
Stinson						
Rise						
Perpetua						

Table A.2: Effects of filters with different α values and intensity $\beta = 1$

Filter	Original	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$	$\alpha = 1.0$
Clarendon						
Gingham						
Juno						
Reyes						
Lark						
Hudson						
Slumber						
Stinson						
Rise						
Perpetua						

Appendix B

Ablation studies

B.1 Per-instance Single Objective attack

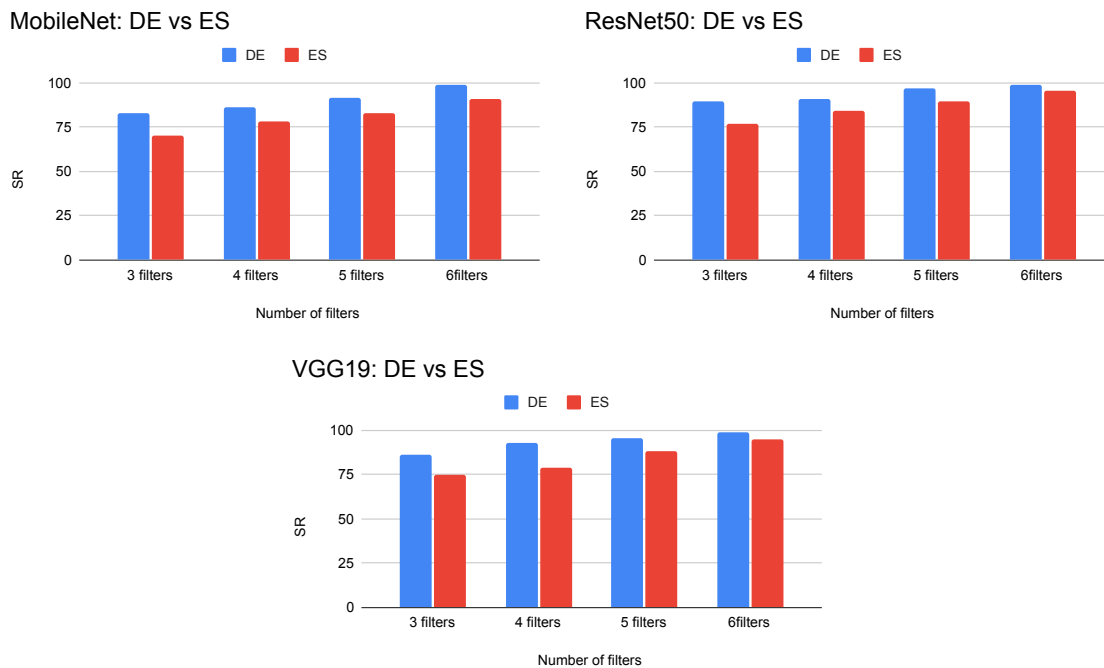


Figure B.1: Success Rate on MobileNet, ResNet50, VGG19 with DE and ES.

We compare the results of the attacks on ImageNet classifier when using DE and ES as inner optimization algorithms. For this analysis, we randomly sampled 100 images from the ImageNet validation dataset. We use the following hyperparameters for the outer optimization

set: population size $N_{out} = 10$, number of generations $G_{out} = 10$, and mutation probability $\rho = 0.5$. We configure DE with $F = 0.7$, $CR=0.5$, number of generations = 3 and population size = 5. We configured ES with a population size $N_{in} = 5$ and number of generation $G_{in} = 3$, an initial learning rate = 0.1 and decay rate =0.75.

Overall, DE reaches higher success rate. The difference between ES and DE is the biggest when using 3 filters. However, when using 6 filters, the gap between the two is not significant on 2 out of 3 models. It is important to note that the computation time of DE is $\approx 3\times$ longer than ES. We do not consider the gain in performance obtained by DE to be worth the extra computation time. Thus, we chose ES since it has the best trade-off between time and success rate.

B.2 Per-instance Multi-objective attack: Emotion Recognition

Ablation study: We conduct ablation experiments to verify the contribution of the inner optimization step implemented with ES and DE. First, we substitute the inner optimization step with a random approach: in this case the values of the filters’ parameters are changed randomly. We call this setup *Inner-random*. Then, we completely remove the inner step and only run the outer algorithm on the sequence of filters with two configuration: 1) keeping the values of parameters fixed with default values during mutation (denoted as *Outer-default*); 2) allowing the mutation to randomly change the parameters of the substituting filter (denoted as *Outer-random*). We use the same hyperparameters setup as in the previous experiments, that is an outer population size =10, number of generations = 10 and mutation probability $\rho = 0.5$. In Table B.1 we report the success rates obtain for each experiment and compare it with the results obtain with *Inner-ES* and *Inner-DE*.

Across the board, we notice an increase in Success Rate when using more sophisticated parameters optimization, such as ES and DE, with the biggest improvement when using only 3 filters, which is the most difficult scenario given that the adversarial perturbations has to be found using less image manipulation operations. These results demonstrate that the adversarial optimization algorithm benefits from the inner step. We also present the

Filters	Outer-default	Outer-random	Inner-random	Inner-ES	Inner-DE
3 filters	85.00	86.25	80.00	91.25	90.00
4 filters	90.00	92.50	92.50	93.75	92.50
5 filters	95.00	91.25	96.25	96.25	100.00

Table B.1: Success Rate (SR%) for different experiment setup.

Optimization	3 filters	4 filters	5 filters
Inner-ES	18s	20s	22s
Inner-DE	65s	70s	75s

Table B.2: Average time (seconds) for one outer generation for ES and DE. For *Outer-default*, *Outer-random*, *Inner-random* the average time per generation is approximately 3 seconds.

distribution of SSIM scores (Figure B.2) for each experimental setup. We observe that *Inner-DE* generates adversarial images with the highest similarity index. Even though there is no considerable difference between the SSIM score of *Inner-ES* and the ablated variants, *Inner-ES* achieves higher success rate.

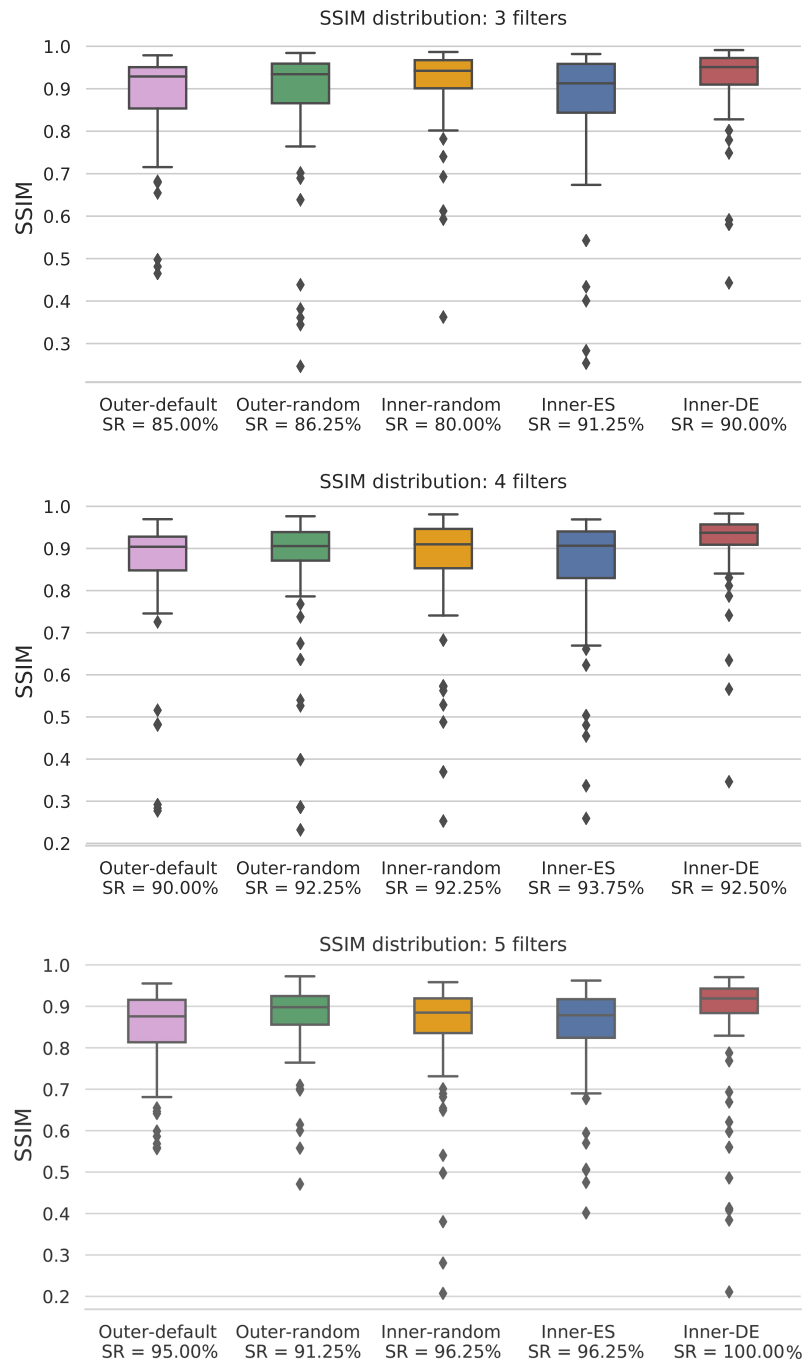


Figure B.2: SSIM score distribution for different experiments and filters.

Appendix C

Case study: Many-objective attack

C.1 From multi to many-objective problem

We extend the problem of multi-objective universal attack, that considers only two objective functions, to a many-objective attack, where three objectives are taken into account: the success rate, detection rate and a measure to control the amount of the applied perturbation. In this context, we formulate the optimization problem as:

$$\text{minimize } \mathcal{F}(X, X^*) = \{1.0 - SR(X, X^*), DR(X, X^*), PA(X^*)\} \quad (\text{C.1})$$

where SR is the success rate defined in Equation 4.15, DR is the detection rate defined in Equation 4.16 and PA is the applied perturbation defined as:

$$PA(X^*) = \frac{1}{n} \sum_{i=0}^n MSE(x_i^*, A(x_i^*)), \quad (\text{C.2})$$

where A is a U-net convolutional autoencoder model trained on the original dataset X to automatically control the perturbations applied by means of the reconstruction error computed using the mean squared error (MSE) between the perturbed images x_i^* and its reconstructed variant. The selection process is handled by the NSGA-III [84] survival technique, which is an extension of the NSGA-II to problems dealing with three objectives.

Experimental setup: We use the Carlini CNN model as target network, the CIFAR-10

dataset, and Feature Squeezing as detection method as in the previous experiment of Section 4.3.3. As inner optimizer we choose ES since is the best performing one in the universal multi-objective attack. To measure the adversarial perturbation we use an anomaly detection approach where we consider as anomalous such images, that after the modification, deviate from the distribution generated by the clean images. We adopt a U-net autoencoder to automatically compute the amount of perturbation applied to the images by means of its reconstruction error. By training the U-net model to minimize the reconstruction error defined by the Mean Squared Error (MSE) on the clean set of images, we can use the MSE to evaluate how much an image was altered: images heavily modified by the filters will have a higher reconstruction error than the clean images. During the optimization process we aim at finding the optimal filter configuration that results in a small reconstruction error.





No. of Filters	Successful Adversarial Examples
5	
6	
7	
8	
Label names	0:airplane, 1:automobile, 2:bird, 3:cat, 4:deer, 5:dog, 6:frog, 7:horse, 8:ship, 9:truck

Table C.1: Some successful adversarial attacks showing dull and sandblasted effects.

Results: For the attacks, we use 5,6,7, and 8 image filters. First of all we analyzed the Attack Success Rate and the Detection Rate obtained with sequences of filters of different lengths. We computed the SR and DR on both training and test set in order to evaluate also the generalization ability of the algorithm. The results are reported in Table C.2. Also in this case, increasing the number of filters corresponds to an improvement of the SR values, resulting in a slight deterioration of the quality of the image. It is interesting to note that, in

all the cases, the detection rate is very low. This indicates that our attack is highly effective since less than 5% of the successful adversarial examples have been identified as illegitimate. Furthermore, we obtain a very good generalization ability given that we only lose between 2% and 6% on SR while still keeping the DR extremely low. With respect to the quality of the images some considerations have to be made. In all the cases images do not present artifacts like the ones produced by other [23,87] that, in general, can be easily detected by the majority of the detection systems [38]. This is clearly an advantage produced by the use of image filters instead of changing single pixels or adding textures [26,87]. On the other hand, the composition of more filters sometimes can produce dull and sandblasted effects that are not sufficiently recognized by the U-net, as shown in Table C.1. Thus, further investigation is necessary in order to find a more suitable method to assess the image quality.

No. of Filters	Training set		Test set	
	SR %	DR %	SR %	DR %
5	74.00	4.00	71.87	2.44
6	80.50	1.20	78.59	1.92
7	82.50	2.40	79.51	2.33
8	83.50	1.19	79.18	1.21

Table C.2: Attack success rate (SR %) and Detection Rate (DR %).

Bibliography

- [1] I. Krizhevsky, A. Sutskever and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Adv. Neural Inf. Process. Syst., 2012.
- [2] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In Proc. Int. Conf. on Learning Represent., 2015.
- [3] K. He, S. Zhang, X. Ren, and J. Sun. Deep residual learning for image recognition. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016.
- [4] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861v1 [cs.CV], 2017.
- [5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In Proc. Int. Conf. on Learning Represent., 2021.
- [6] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Adv. Neural Inf. Process. Syst., 2015.
- [7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016.
- [8] T.Y Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017.

- [9] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In Proc. Eur. Conf. Comput. Vis., 2020.
- [10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In Proc. Eur. Conf. Comput. Vis., 2016.
- [11] B. Cheng, I. Misra, A. G Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022.
- [12] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015.
- [13] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In Int. Conf. Medical Image Comput. and Computer-Assisted Intervention, 2015.
- [14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In Proc. IEEE Int. Conf. Comput. Vis., 2017.
- [15] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. In IEEE Trans. Pattern Anal. Mach. Intell., 2017.
- [16] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. arXiv:1312.6199v4 [cs.CV], 2014.
- [17] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In Proc. Int. Conf. on Learning Represent., 2015.
- [18] A. Kurakin, I.J. Goodfellow, and S. Bengio. Adversarial machine learning at scale. In Proc. Int. Conf. on Learning Represent., 2017.
- [19] A. Kurakin, Ian J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In Int. Joint Conf. on Artificial Intell., 2017.

- [20] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In IEEE Eur. Symp. on Security and Privacy, 2016.
- [21] Nicolas Papernot, P. McDaniel, Ian J. Goodfellow, S. Jha, Z. Y. Celik, and A. Swami. Practical black-box attacks against machine learning. Proc. ACM Asia Conf. Computer Commun. Security, 2017.
- [22] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In Proc. Int. Conf. on Learning Represent., 2018.
- [23] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In IEEE Symp. on Security and Privacy, 2017.
- [24] X. Wei, S. Liang, N. Chen, and X. Cao. Transferable adversarial attacks for image and video object detection. In Int. Joint Conf. on Artificial Intell., 2018.
- [25] S.M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016.
- [26] A. Bhattad, M. J. Chong, K. Liang, B. Li, and D. A. Forsyth. Unrestricted adversarial examples via semantic manipulation. In Proc. Int. Conf. on Learning Represent., 2020.
- [27] A. S. Shamsabadi, C. Oh, and A. Cavallaro. Edgefool: an adversarial image enhancement filter. In Proc. IEEE Int. Conf. Acoustics, Speech Signal Process., 2019.
- [28] H. Hosseini and R. Poovendran. Semantic adversarial examples. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018.
- [29] Z. Zhao, Z. Liu, and M. A. Larson. Adversarial robustness against image color transformation within parametric filter space. arXiv:2011.06690v2 [cs.CV], 2020.
- [30] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin. Black-box adversarial attacks with limited queries and information. In Int. Conf. on Mach. Learning, 2018.

- [31] A.S. Shamsabadi, R. Sanchez-Matilla, and A. Cavallaro. ColorFool: Semantic adversarial colorization. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020.
- [32] Y. Dong, F. Liao, H. Pang, T. and Su, J. Zhu, and J. Hu, X. and Li. Boosting adversarial attacks with momentum. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018.
- [33] W. Brendel, J. Rauber, and M. Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In Proc. Int. Conf. on Learning Represent., 2018.
- [34] M. Cheng, S. Singh, P. H. Chen, P.Y. Chen, S. Liu, and C.-J. Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. In Proc. Int. Conf. on Learning Represent., 2020.
- [35] C. Xie, Z. Zhang, J. Wang, Y. Zhou, Z. Ren, and A.L. Yuille. Improving transferability of adversarial examples with input diversity. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019.
- [36] D. Wu, Y. Wang, S.T. Xia, J. Bailey, and X. Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. In Proc. Int. Conf. on Learning Represent., 2020.
- [37] Y. Dong, T. Pang, H. Su, and J. Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019.
- [38] W. Xu, D. Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In Network and Distr. Systems Security Symp., 2018.
- [39] G. Dziugaite, Z. Ghahramani, and D. M. Roy. A study of the effect of jpg compression on adversarial images. arXiv:1608.00853v1 [cs.CV], 2016.
- [40] A. Van Looveren, G. Vacanti, J. Klaise, A. Coca, and O. Cobb. Alibi detect: Algorithms for outlier, adversarial and drift detection. <https://github.com/SeldonIO/alibi-detect>, 2019.

- [41] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In IEEE Symp. on Security and Privacy, 2016.
- [42] D. Arcelli, A. E. Baia, A. Milani, and V. Poggioni. Enhance while protecting: privacy preserving image filtering. In Proc of IEEE/WIC/ACM Int. Joint Conf. on Web Intell. and Intell. Agent Technology, 2021.
- [43] S.M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017.
- [44] J. Hayes and G. Danezis. Learning universal adversarial perturbations with generative models. In IEEE Security and Privacy Workshops, 2018.
- [45] K. R. Mopuri, A. Ganeshan, and R. V. Babu. Generalizable data-free objective for crafting universal adversarial perturbations. IEEE Trans. Pattern Anal. Mach. Intell., 2018.
- [46] K.R. Mopuri, P.K. Uppala, and V.R Babu. Ask, acquire, and attack: Data-free uap generation using class impressions. In Proc. Eur. Conf. Comput. Vis., 2018.
- [47] A. E. Baia, G. Di Bari, and V. Poggioni. Effective universal unrestricted adversarial attacks using a moe approach. In Proc. Int. Conf. Appl. of Evolutionary Comput., 2021.
- [48] P. Chen, H. Zhang, Y. Sharma, J. Yi, and C.J. Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In ACM Workshop on Artificial Intell. and Security, 2017.
- [49] N. Narodytska and S. Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In Conf. Comput. Vis. Pattern Recognit. Workshops, 2017.
- [50] M. Alzantot, Y. Sharma, S. Chakraborty, H. Zhang, C.J. Hsieh, and M.B. Srivastava. Genattack. In Proc. of the Genetic and Evolutionary Comput. Conf., 2019.

- [51] R. Mosli, M. Wright, B. Yuan, and Y. Pan. They might not be giants: Crafting black-box adversarial examples with fewer queries using particle swarm optimization. arXiv:1909.07490v1 [cs.LG], 2019.
- [52] P. Vidnerová and R. Neruda. Vulnerability of classifiers to evolutionary generated adversarial examples. Neural networks, J. of the Int. Neural Network Society, 2020.
- [53] J. Su, D. V. Vargas, and K. Sakurai. One pixel attack for fooling deep neural networks. IEEE Trans. on Evolutionary Comput., 2019.
- [54] T. Suzuki, S. Takeshita, and S. Ono. Adversarial example generation using evolutionary multi-objective optimization. In IEEE Congress on Evolutionary Comput., 2019.
- [55] Y. Deng and X. Zhang, C. and Wang. A multi-objective examples generation approach to fool the deep neural networks in the black-box scenario. In IEEE Int. Conf. on Data Science in Cyberspace, 2019.
- [56] X. Li, H. and Xu, X. Zhang, S. Yang, and B. Li. Qeba: Query-efficient boundary-based blackbox attack. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020.
- [57] C. Oh, A. Xompero, and A. Cavallaro. Chapter 15 - Visual adversarial attacks and defenses. In Advanced Methods and Deep Learning in Computer Vision. Academic Press, 2022.
- [58] X. Liu, H. Yang, Z. Liu, L. Song, Y. Chen, and H. Li. Dpatch: An adversarial patch attack on object detectors. In Proc. AAAI Workshop on Artificial Intell. Safety, 2019.
- [59] K. T. Co, L. Muñoz-González, S. de Maupeou, and E. C. Lupu. Procedural noise adversarial examples for black-box attacks on deep convolutional networks. In Proc. ACM SIGSAC Conf. Computer Commun. Security, 2019.
- [60] L. Wang. A survey on IQA. arXiv:2109.00347v2 [eess.IV], 2021.
- [61] S. Xu, S. Jiang, and W. Min. No-reference/blind image quality assessment: A survey. IETE Technical Review, 2017.

- [62] G. Zhai and X. Min. Perceptual image quality assessment: a survey. Science China Information Sciences, 2020.
- [63] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process., 2004.
- [64] H. Talebi and P. Milanfar. Nima: Neural image assessment. IEEE Trans. Image Process., 2018.
- [65] A. Mittal, R. Soundararajan, and A.C. Bovik. Making a “completely blind” image quality analyzer. IEEE Signal Processing Letters, 2013.
- [66] S. Yang, T. Wu, S. Shi, S. Lao, Y. Gong, M. Cao, J. Wang, and Y. Yang. Maniqa: Multi-dimension attention network for no-reference image quality assessment. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022.
- [67] N. Murray, L. Marchesotti, and F. Perronnin. Ava: A large-scale database for aesthetic visual analysis. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2012.
- [68] J. Gu, H. Cai, C. Dong, J. S Ren, R. Timofte, Y. Gong, S. Lao, S. Shi, J. Wang, S. Yang, et al. NTIRE 2022 challenge on perceptual image quality assessment. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022.
- [69] J.Gu, H. Cai, H. Chen, X. Ye, J. Ren, and C. Dong. Pipal: a large-scale image quality assessment dataset for perceptual image restoration. In Proc. Eur. Conf. Comput. Vis., 2020.
- [70] Y. Wu, X. Wang, G.Li, and Y. Shan. AnimeSR: Learning real-world super-resolution models for animation videos. In Adv. Neural Inf. Process. Syst., 2022.
- [71] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. D. McDaniel. Ensemble adversarial training: Attacks and defenses. In Proc. Int. Conf. on Learning Represent., 2018.
- [72] N. Akhtar, J. Liu, and A. Mian. Defense against universal adversarial perturbations. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018.

- [73] F. Kinli, B. Ozcan, and F. Kirac. Instagram filter removal on fashionable images. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021.
- [74] T. Song, Y. Kim, S. Nowozin, S. Ermon, and N. Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In Proc. Int. Conf. on Learning Represent., 2018.
- [75] S. Gu and L. Rigazio. Towards deep neural network architectures robust to adversarial examples. In Proc. Int. Conf. on Learning Represent., 2015.
- [76] D. Meng and H. Chen. Magnet: A two-pronged defense against adversarial examples. In Proc. ACM SIGSAC Conf. Computer Commun. Security, 2017.
- [77] Z. Michalewicz. Genetic algorithms + data structures = evolution programs. Springer-Verlag, 1992.
- [78] R. Storn and K. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. of Global Optimization, 1997.
- [79] I. Rechenberg. Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution. Dr.-Ing. Thesis, Technical University of Berlin, Department of Process Engineering, 1971.
- [80] B. Leite, A. O. S. da Costa, and E. F. da Costa Junior. Multi-objective optimization of adiabatic styrene reactors using generalized differential evolution 3 (gde3). Chemical Engineering Science, 2023.
- [81] B. V. Babu and B. Anbarasu. Multi-objective differential evolution (mode): An evolutionary algorithm for multi-objective optimization problems (moops). i-manager's J. on Future Engineering and Technology, 2007.
- [82] T. Salimans, J. Ho, S. Chen, X. Sidor, and I. Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. arXiv:1703.03864v2 [stat.ML], 2017.
- [83] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans. on Evolutionary Comput., 2002.

- [84] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. IEEE Trans. on Evolutionary Comput., 2014.
- [85] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In Proc. Int. Conf. on Learning Represent., 2018.
- [86] Y. Liu, X. Chen, C. Liu, and D. Song. Delving into transferable adversarial examples and black-box attacks. In Proc. Int. Conf. on Learning Represent., 2017.
- [87] X. Yuan, P. He, Q. Zhu, and X. Li. Adversarial examples: Attacks and defenses for deep learning. IEEE Trans. on Neural Networks and Learning Systems, 2019.
- [88] C. Xie, Y. Wu, L. van der Maaten, A. L. Yuille, and K. He. Feature denoising for improving adversarial robustness. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019.
- [89] A. Sen, X. Zhu, L. Marshall, and R. Nowak. Should adversarial attacks use pixel p-norm? arXiv:1906.02439v1 [cs.LG], 2019.
- [90] C. Kanbak, S.M. Moosavi-Dezfooli, and P. Frossard. Geometric robustness of deep networks: analysis and improvement. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018.
- [91] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry. Exploring the landscape of spatial robustness. In Int. Conf. on Mach. Learning, 2019.
- [92] M. Sharif, S. Bhagavatula, L. Bauer, and M.K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In Proc. ACM SIGSAC Conf. Computer Commun. Security, 2016.
- [93] A. Joshi, A. Mukherjee, S. Sarkar, and C. Hegde. Semantic adversarial attacks: Parametric transformations that fool deep classifiers. In Proc. IEEE Int. Conf. Comput. Vis., 2019.

- [94] A. Marchisio, G. Caramia, M. Martina, and M. Shafique. fakeweather: Adversarial attacks for deep neural networks emulating weather conditions on the camera lens of autonomous systems. In Int. Joint Conf. on Neural Networks, 2022.
- [95] C. Laidlaw and S. Feizi. Functional adversarial attacks. In Adv. Neural Inf. Process. Syst., 2019.
- [96] A.S. Shamsabadi, C. Oh, and A. Cavallaro. Semantically adversarial learnable filters. IEEE Trans. Image Process., 2021.
- [97] Z. Zhao, Z. Liu, and M. Larson. Adversarial color enhancement: Generating unrestricted adversarial images by optimizing a color filter. In Proc. Brit. Mach. Vis. Conf., 2020.
- [98] L. Sun, F. Juefei-Xu, Y. Huang, Q. Guo, J. Zhu, J. Feng, Y. Liu, and G. Pu. Ala: Adversarial lightness attack via naturalness-aware regularizations. arXiv:2201.06070v1 [cs.CV], 2022.
- [99] Y. Sun, C. Wu, K. Zheng, and X. Niu. Adv-emotion: The facial expression adversarial attack. Int. J. of Pattern Recognit. and Artificial Intell., 2021.
- [100] Y. Sun, J. Yin, C. Wu, K. Zheng, and X. Niu. Generating facial expression adversarial examples based on saliency map. Image and Vis. Comput., 2021.
- [101] K.R. Mopuri, U. Ojha, U. Garg, and R.V. Babu. Nag: Network for adversary generation. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018.
- [102] O. Poursaeed, I. Katsman, B. Gao, and S. Belongie. Generative adversarial perturbations. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018.
- [103] Z. Wang, Y. Yang, J. Li, and X. Zhu. Universal adversarial perturbations generative network. World Wide Web, Internet and Web Inf. Syst., 2022.
- [104] J. Wang, Z. Yin, J. Jiang, and Y. Du. Attention-guided black-box adversarial attacks with large-scale multiobjective evolutionary optimization. Int. J. of Intell. Syst., 2022.

- [105] C. Xie, J. Wang, Z. Zhang, Y. Zhou, and A. Xie, L. and Yuille. Adversarial examples for semantic segmentation and object detection. In Proc. IEEE Int. Conf. Comput. Vis., 2017.
- [106] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. arXiv:1804.02767v1 [cs.CV], 2018.
- [107] T.Y. Lin, M. Maire, S.J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick. Microsoft coco: Common objects in context. In Proc. Eur. Conf. Comput. Vis., 2014.
- [108] Y. Wang, Y. Tan, W. Zhang, Y. Zhao, and X. Kuang. An adversarial attack on dnn-based black-box object detectors. J. of Network and Computer Appl., 2020.
- [109] Y. Li, D. Tian, M.C. Chang, X. Bian, and S. Lyu. Robust adversarial perturbation on deep proposal-based models. In Proc. Brit. Mach. Vis. Conf., 2018.
- [110] S. Liang, B. Wu, Y. Fan, and X. Wei, X. and Cao. Parallel rectangle flip attack: A query-based black-box attack against object detection. In Proc. IEEE Int. Conf. Comput. Vis., 2021.
- [111] Y. Lu, Y. Jia, J. Wang, B. Li, W. Chai, L. Carin, and S. Velipasalar. Enhancing cross-task black-box transferability of adversarial examples with dispersion reduction. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020.
- [112] A.J. Bose and P. Aarabi. Adversarial attacks on face detectors using neural net based constrained optimization. In Int. Workshop on Multimedia Signal Process., 2018.
- [113] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barredo, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information Fusion, 2020.
- [114] S.S.Y. Kim, E.A. Watkins, O. Russakovsky, R.C. Fong, and A. Monroy-Hernández. "Help Me Help the AI": Understanding how explainability can support human-ai interaction. In Conf. on Human Factors in Comput. Syst., 2022.

- [115] A. Adadi and M. Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). IEEE access, 2018.
- [116] A. Arias-Duart, F. Par’es, D. García-Gasulla, and V. Giménez-Ábalos. Focus! rating xai methods and finding biases. In IEEE Int. Conf. on Fuzzy Syst., 2021.
- [117] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proc. IEEE Int. Conf. Comput. Vis., 2017.
- [118] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In Proc. Int. Conf. on Learning Represent., 2014.
- [119] M.T. Ribeiro, S. Singh, and C. Guestrin. ” Why Should I Trust You? ”: Explaining the predictions of any classifier. In Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016.
- [120] S. M. Lundberg and S.I Lee. A unified approach to interpreting model predictions. In Int. Conf. on Neural Inf. Process. Syst., 2017.
- [121] L.A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell. Generating visual explanations. In Proc. Eur. Conf. Comput. Vis., 2016.
- [122] M. Kayser, O.M Camburu, L. Salewski, C. Emde, V. Do, Z. Akata, and T. Lukasiewicz. e-vil: A dataset and benchmark for natural language explanations in vision-language tasks. In Proc. IEEE Int. Conf. Comput. Vis., 2021.
- [123] A. Marasović, C. Bhagavatula, J.S. Park, R. Le Bras, N. A. Smith, and Y. Choi. Natural language rationales with full-stack visual reasoning: From pixels to semantic frames to commonsense graphs. In Findings ACL : EMNLP 2020, 2020.
- [124] Q. Li, Q. Tao, S. Joty, J. Cai, and J. Luo. Vqa-e: Explaining, elaborating, and enhancing your answers for visual questions. In Proc. Eur. Conf. Comput. Vis., 2018.

- [125] V. Do, O.M. Camburu, Z. Akata, and T. Lukasiewicz. e-snli-ve: Corrected visual-textual entailment with natural language explanations. arXiv:2004.03744 [cs.CL], 2020.
- [126] F. Sammani, T. Mukherjee, and N. Deligiannis. Nlx-gpt: A model for natural language explanations in vision and vision-language tasks. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022.
- [127] D. H. Park, L. A. Hendricks, Z. Akata, A. Rohrbach, B. Schiele, T. Darrell, and M. Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018.
- [128] J. Wu and R. Mooney. Faithful multimodal explanation for visual question answering. In Proc. ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, 2019.
- [129] A. Ghorbani, A. Abid, and J. Zou. Interpretation of neural networks is fragile. In Proc. AAAI Conf. on Artificial Intell., 2019.
- [130] A.K. Dombrowski, M. Alber, C. Anders, M. Ackermann, K.R Müller, and P. Kessel. Explanations can be manipulated and geometry is to blame. In Adv. Neural Inf. Process. Syst., 2019.
- [131] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In Proc. AAAI/ACM Conf. on AI, Ethics, and Society, 2020.
- [132] X. Xu, X. Chen, C. Liu, A. Rohrbach, T. Darrell, and D. Song. Fooling vision and language models despite localization and attention mechanism. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018.
- [133] H. Chen, H. Zhang, P.Y Chen, J. Yi, and C.J Hsieh. Attacking visual language grounding with adversarial examples: A case study on neural image captioning. In Proc. of the Annual Meeting ACL, 2018.

- [134] Y. Xu, B. Wu, F. Shen, Y. Fan, Y. Zhang, H. Tao Shen, and W. Liu. Exact adversarial attack to image captioning via structured output learning with latent variables. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019.
- [135] S. Zhang, Z. Wang, X. Xu, X. Guan, and Y. Yang. Fooled by imagination: Adversarial attack to image captioning via perturbation in complex domain. In Proc. IEEE Int. Conf. on Multimedia and Expo, 2020.
- [136] H. Wu, Y. Liu, H. Cai, and S. He. Learning transferable perturbations for image captioning. ACM Trans. Multimedia Comput. Commun. Appl., 2022.
- [137] H. Kwon and S. Kim. Restricted-area adversarial example attack for image captioning model. Wireless Commun. and Mobile Comput., 2022.
- [138] J. Ji, X. Sun, Y. Zhou, R. Ji, F. Chen, J. Liu, and Q. Tian. Attacking image captioning towards accuracy-preserving target words removal. In Proc. ACM Int. Conf. Multimedia, 2020.
- [139] V. Sharma, A. Kalra, S.C. Vaibhav, L. Patel, and L.P. Morency. Attend and attack: Attention guided adversarial attacks on visual question answering models. In Adv. Neural Inf. Process. Syst., 2018.
- [140] A. Chaturvedi and U. Garain. Mimic and fool: A task-agnostic adversarial attack. IEEE Trans. on Neural Networks and Learning Syst., 2020.
- [141] K. Yang, W. Lin, M. Barman, F. Condessa, and Z. Kolter. Defending multimodal fusion models against single-source adversaries. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021.
- [142] J. Zhang, Q. Yi, and J. Sang. Towards adversarial attack on vision-language pre-training models. In Proc. ACM Int. Conf. Multimedia, 2022.
- [143] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2009.

- [144] M. Sandler, A. Howard, M. Zhu, and L.C. Zhmoginov, A.and Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018.
- [145] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016.
- [146] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556v6 [cs.CV], 2014.
- [147] K. R. Mopuri, V. Shaj, and R.V. Babu. Adversarial fooling beyond "flipping the label". In Conf. Comput. Vis. Pattern Recognit. Workshops, 2020.
- [148] A. Ganeshan, B.S. Vivek, and R. V. Babu. Fda: Feature disruptive attack. In Proc. IEEE Int. Conf. Comput. Vis., 2019.
- [149] R.W. Picard. Affective computing: Challenges. Int. J. Human-Computer Studies, 2003.
- [150] P. Ekman. An argument for basic emotions. Cognition and Emotion, 1992.
- [151] S. Generosi, A.and Ceccacci and M. Mengoni. A deep learning-based system to track and analyze customer behavior in retail store. In IEEE Int. Conf. on Consumer Electronics, 2018.
- [152] A. Gorrini, L. Crociani, G. Vizzari, and S. Bandini. Stress estimation in pedestrian crowds: Experimental data and simulations results. Web Intelligence, 2019.
- [153] Z. Xing, Y.and Hu, Z. Huang, C. Lv, D. Cao, and E. Velenis. Multi-scale driver behaviors reasoning system for intelligent vehicles based on a joint deep learning framework. In IEEE Int. Conf. on Syst., Man, and Cybernetics, 2020.
- [154] E. Ferrara and Z. Yang. Quantifying the effect of sentiment on information diffusion in social media. PeerJ Computer Science, 2015.
- [155] F. D’Errico and I. Poggi. "Humble" politicians and their multimodal communication. In Int. Conf. Comput. Science and Its Applications, 2017.

- [156] J. Carpenter. The Quiet Professional: An investigation of US military Explosive Ordnance Disposal personnel interactions with everyday field robots. PhD thesis, University of Washington, 2013.
- [157] P. Ekman and W.V. Friesen. A new pan-cultural facial expression of emotion. Motivation and Emotion, 1986.
- [158] A. Mollahosseini, B. Hasani, and M.H. Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. IEEE Trans. on Affective Comput., 2019.
- [159] M. Guarnera, Z. Hichy, M. Cascio, S. Carrubba, and S. Buccheri. Facial expressions and the ability to recognize emotions from the eyes or mouth: A comparison between children and adults. The Journal of Genetic Psychology, 2017.
- [160] M. Guarnera, Z. Hichy, M. Cascio, and S. Carrubba. Facial expressions and ability to recognize emotions from eyes or mouth in children. Europe’s journal of psychology, 2015.
- [161] M. Guarnera, P. Magnano, M. Pellerone, M. an Cascio, V. Squatrito, and S. Buccheri. Facial expressions and the ability to recognize emotions from the eyes or mouth: A comparison among old adults, young adults, and children. The Journal of genetic psychology, 2018.
- [162] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer. A review and comparative study on probabilistic object detection in autonomous driving. IEEE Intell. Transportation Syst., 2021.
- [163] X. Chen, X. Wang, K. Zhang, K.M. Fung, T.C. Thai, K. Moore, R.S. Mannel, H. Liu, B. Zheng, and Y. Qiu. Recent advances and clinical applications of deep learning in medical image analysis. Medical Image Analysis, 2022.
- [164] I. Ahmed, G. Jeon, A. Chehri, and M.M. Hassan. Adapting gaussian yolov3 with transfer learning for overhead view human detection in smart cities and societies. Sustainable Cities and Society, 2021.

- [165] H. Bae, J. Jang, D. Jung, H. Jang, H. and Ha, H. Lee, and S. Yoon. Security and privacy issues in deep learning. arXiv:1807.11655v4 [cs.CR], 2018.
- [166] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In Proc. ACM SIGSAC Conf. Computer Commun. Security, 2015.
- [167] F. Mireshghallah, M. Taram, P. Vepakomma, R. Singh, A. and Raskar, and H. Esmaeilzadeh. Privacy in deep learning: A survey. arXiv:2004.12254v5 [cs.LG], 2020.
- [168] A. Bochkovskiy, C.Y. Wang, and H.Y.M. Liao. Yolov4: Optimal speed and accuracy of object detection. arXiv:2004.10934v1 [cs.CV], 2020.
- [169] A. Wang, C.Y. and Bochkovskiy and H.Y.M. Liao. Scaled-yolov4: Scaling cross stage partial network. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021.
- [170] S. Wiegrefe, A. Marasović, and N.A. Smith. Measuring association between labels and free-text rationales. In Conf. on Empirical Methods in Natural Language Process., 2021.
- [171] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Conf. on Empirical Methods in Natural Language Process., 2019.
- [172] M.A. Cliniciu, A. Eshghi, and H. Hastie. A study of automatic metrics for the evaluation of natural language explanations. In Conf. of the Eur. Chapter of the ACL: Main Volume, 2021.
- [173] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In Int. Conf. on Mach. Learning, 2021.
- [174] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In Adv. Neural Inf. Process. Syst. EMC² Workshop, 2019.

- [175] B.A. Plummer, L. Wang, C.M. Cervantes, J.C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In Proc. IEEE Int. Conf. Comput. Vis., 2015.
- [176] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.J. Li, D.A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vis., 2017.
- [177] J. Krause, J. Johnson, R. Krishna, and L. Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017.
- [178] T. Lei, P. Liu, X. Jia, X. Zhang, H. Meng, and A.K. Nandi. Automatic fuzzy clustering framework for image segmentation. IEEE Trans. on Fuzzy Syst., 2020.
- [179] D.Hasler and S. Süsstrunk. Human vision and electronic imaging - measuring colorfulness in natural images. In SPIE Proceedings, 2003.
- [180] S. Banerjee and A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proc. ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, 2005.