

Evasion Attacks Against On-Device Violent Image Classification Deep Learning Models

Anton Shumilin

Evasion Attacks Against On-Device Violent Image Classifi- cation Deep Learning Models

Anton Shumilin

Thesis submitted in partial fulfillment of the requirements for
the degree of Bachelor of Science in Technology.
Otaniemi, 12 Dec 2025

Supervisor: associate professor Maarit Korpi-Lagg
Advisor: assistant professor Sebastian Szyller

Aalto University
School of Science
Bachelor's Programme in Science and Technology

Author

Anton Shumilin

Title

Evasion Attacks Against Violent On-Device Image Classification Deep Learning Models

School School of Science**Degree programme** Bachelor's Programme in Science and Technology**Major** Data Science**Code** SCI3095**Supervisor** associate professor Maarit Korpi-Lagg**Advisor** assistant professor Sebastian Szyller**Level** Bachelor's thesis **Date** 12 Dec 2025 **Pages** 38 **Language** English**Abstract**

Deep learning models can be effectively utilized in many applications, including the detection of violent images. Unfortunately, these models can be vulnerable to attacks that introduce imperceptible modifications to the image. Such attacks can cause misclassification, which may lead to inefficiencies in public safety and to the spread of violent content. However, despite the risks, the research comparing attacks on violence detectors is limited.

This thesis explores a range of attacks applicable to the on-device violence classification task. It presents a literature review that identifies various types of attacks under the threat model tailored to the task and proposes a taxonomy of the attack methods based on their scenarios and generation principles. The review complements the taxonomy with the analysis of the attack differences and recent improvements. The review is supplemented by an experiment, which evaluates a subset of the discussed attacks on lightweight violence classification models. The experiment demonstrates a significant vulnerability of undefended models and illustrates the effect of various attack constraints on the imperceptibility and generation time.

Keywords Evasion attacks, computer vision, deep learning, image classification, violence detection**urn** <https://aaltodoc.aalto.fi>

Contents

Abstract	ii
Contents	iii
1. Introduction	1
2. Background	3
2.1 Preliminaries	3
2.2 Threat Model	5
2.2.1 Degree of Knowledge	5
2.2.2 Adversarial Capabilities	6
2.2.3 Adversary Goals	6
2.3 Attack Selection	7
3. White-box Attacks	8
3.1 Norm-constrained Optimization	9
3.2 Generative Models	13
3.3 Other Constraints	15
4. Black and Gray-box Attacks	18
4.1 Transfer-based	19
4.2 Processing-based	20
4.3 Query-based	21
5. Experiment	23
5.1 Methodology	23
5.2 Metrics	24
5.3 Results	26
6. Conclusion	29

1. Introduction

Deep learning models have been utilized successfully in computer vision tasks for more than a decade [1] and have found many applications in the real world, from skin cancer analysis [2] to traffic sign recognition [3]. One of the tasks that could benefit from a machine learning approach is automated detection of violent images. It can be useful in various areas, such as content moderation to prevent the spread of media that violate moderation guidelines, and public safety to facilitate timely detection of threats and conflicts.

Violence detection may be needed in cases where model inference on a remote server is risky or infeasible. For example, the camera feed may contain sensitive information, or the system may not have a reliable Internet connection. With the advancement of computational capabilities of mobile and edge devices and the development of lightweight architectures, AI inference closer to the data source, or on the edge, is becoming increasingly popular [4]. Following these developments, the violence detection model can be deployed directly on the device.

Despite the general high accuracy achieved by modern image classification models, these models are often vulnerable to *adversarial perturbations*—small changes that are imperceptible or insignificant to humans but which affect model predictions [5]. This instability comes with a range of risks. First, there is a chance of inadvertent mispredictions due to slight changes in the environment, noise, and other factors not covered during model training. Second, models may be vulnerable to evasion attacks, in which the attacker intentionally deceives them to circumvent rules or disrupt system functionality.

A considerable amount of research has been conducted on evasion

attacks in computer vision [6], [7]. However, existing studies either concern a wide category of attacks [6], [8] or do not target on-device models, for example, reviewing attacks on cloud violence detectors [9] instead of on-device attacks. Such general surveys cannot provide an in-depth analysis of a wide range of methods; moreover, some of them overlook recent progress in the field. Thus, the aim of this thesis is to identify the extent of current research on evasion attacks applicable to on-device image violence detection by conducting a state-of-the-art literature review. To supplement the discussion, the thesis provides a practical comparative evaluation of a selection of the attacks. The main contributions of this thesis are the selection and taxonomy of attacks applicable to violence detection, an analysis of recent improvements in those attacks, and an empirical evaluation of the attacks on lightweight deep learning models.

The thesis is divided into six sections. Section 2 provides general information on deep learning models and attacks. Sections 3 and 4 describe the attacks under different settings in detail. Then, Section 5 describes the experiment setup and obtained results. Finally, Section 6 concludes the review and evaluation.

2. Background

This section introduces terms, concepts, and goals required to describe different attack methods. First, it establishes the notation and methods required further, and then more thoroughly describes attack properties.

2.1 Preliminaries

An image classifier is a function $F : \mathcal{S} \rightarrow \{1, \dots, k\}$, where k is the number of classes and $\mathcal{S} \in [0, 1]^{H \times W \times C}$ is a space of images with height H , width W , and number of channels C . This representation can be obtained by resizing the original image and scaling pixel intensities from the standard 0 – 255 range to $[0, 1]$ interval. The image is denoted by x , and the true class associated with it by y .

In the deep learning setting, the classifier is based on a function \mathcal{Z} of the input tensor x and a set of weights θ . \mathcal{Z} returns logits $z_i \in \mathbb{R}$ for each of k classes. The logits can be transformed into confidence scores $f_i \in [0, 1]$ —model certainty of the input belonging to class i —by applying an activation function, such as softmax:

$$f_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}}.$$

Note that binary classification tasks may rely on a single logit to define confidence scores. Subsequently, given the confidence scores, the output of the classifier function F can be computed as $\arg \max_i f_i$.

Confidence scores also define decision boundaries, which are hyperplanes $\pi_{i,j} \subset \mathcal{S}$, such that $\forall x \in \pi_{i,j} : \max_q f_q(x) = f_i(x) = f_j(x)$. These boundaries usually separate the images by the class predicted by the model. In binary classification settings, the indices in

π can be omitted.

Deep learning models are trained by iteratively adjusting the weights θ to minimize a certain loss function $L(\theta, x, y) \rightarrow \mathbb{R}$, which penalizes confidence corresponding to the wrong prediction classes but may also include other terms. A common loss function for classification tasks is cross-entropy, which for a single true class y is

$$L_{CE} = -\log(f_y(\theta, x)).$$

The adversarial objective is to create an adversarial example $\hat{x} \in \mathcal{S}$, such that $\hat{y} = F(\hat{x}) \neq y$. Then, the adversarial perturbation is defined as $\eta = \hat{x} - x$.

Not all adversarial examples represent equally successful attacks. For instance, imperceptibility is a key consideration in an attack, since easily distinguishable adversarial examples can be filtered or not have an intended effect on the human observer. Thus, one important quantifier of an adversarial example is a *distance metric* which represents how different the adversarial example is from the original image. A widely used distance metric is an ℓ_p norm due to its simplicity and computational efficiency. An ℓ_p norm of η is defined as

$$\|\eta\|_p = \left(\sum_i |\eta_i|^p \right)^{1/p}.$$

Some common special cases of an ℓ_p norm include ℓ_∞ , ℓ_0 , and ℓ_2 . ℓ_∞ limits the maximum absolute pixel change to verify if any value deviates significantly from the original one; ℓ_0 identifies the number of changed intensities to constrain the portion of the images that differs from the original; and ℓ_2 is the Euclidean norm of the perturbation, which considers both the number of deviations and their magnitude.

Although the attacks target classification models, generative models may also appear useful. Instead of estimating the probabilities of the target class given an image sample, they estimate the distribution of the images themselves or images with the labels, which allows these models to create new samples not present in the training data.

One type of generative model is a generative adversarial network (GAN). A GAN consists of a generator G , which creates images or other high-dimensional outputs, and a discriminator D , which

detects whether the input has been synthetically generated. The models are trained to find [10]

$$\min_G \max_D \mathcal{L}_{\text{GAN}}(D, G),$$

where $\mathcal{L}_{\text{GAN}}(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} \log D(x) + \mathbb{E}_{z \sim p_z(z)} \log(1 - D(G(z)))$.

Another type of generative model is a diffusion model [11]. Its training is based on adding noise to the original images and then learning to reconstruct them reversing the process.

2.2 Threat Model

This section describes a threat model, which is needed to define the scope and target of the considered attacks. The model covers the following factors: degree of knowledge, adversary capabilities, and adversarial goals [12].

There are two most probable scenarios for evasion attacks against on-device violence image classification models. In the first one, the adversary does not have direct access to the model and therefore can only construct input samples based on the class predicted by the model. This scenario represents the setting where the attacker is a user without knowledge of the internal contents of the application or an indirect user, as in the case of surveillance applications.

However, given that the model is deployed on-device, there is a possibility of the attacker obtaining access to it. Therefore, such a case is also considered in this thesis. Additionally, attacks with full access can serve as the basis for other attack types or be useful to test the robustness of the model under worst-case conditions [12].

2.2.1 Degree of Knowledge

The first scenario is black box [13]—it assumes no access to model parameters, architecture, or any large-scale training dataset. However, the adversary might make an educated guess about the model architecture and parts of the training data by researching widely adopted methods, state-of-the-art solutions, and publicly available datasets. This scenario, in which the attacker does not have access to the model weights but has certain knowledge about the data or

architecture, is referred to as *gray box*.

The second scenario is *white box*, which means that the adversary has access to model weights, architecture, and therefore confidence scores. However, the attacker has no knowledge about the training data and process.

2.2.2 Adversarial Capabilities

The phase of machine learning during which the attack occurs plays the main role in defining adversarial capabilities, i.e., training or an inference phase. Given that this thesis only considers attacks on already deployed models, the scope is solely limited to testing phase attacks. Therefore, an adversary must craft adversarial samples to explore or trick the model without the ability to modify training data, alter the architecture, or otherwise affect the prediction of the model. However, the adversary is not constrained in building their models based on the outputs of the original model and publicly available data.

2.2.3 Adversary Goals

Adversary goals represent the end result that the adversary is trying to achieve. Testing phase attacks can be divided into evasion and exploratory attacks. The former have a goal of causing mispredictions of the targeted model, while the latter is aimed at extracting information about the learning system, such as model architecture, weights, behavior, and training data [14]. In the case of violence detection, the training data and process themselves do not have a significant value to an attacker, unless their goal is building a similar system. Therefore, the considered threat model includes only evasion attacks.

Within this scope, the goals fall into one of three categories [8]: untargeted misclassification, source/target misclassification, and confidence reduction. *Untargeted misclassification* is achieved by making the model predict any class other than the true one. *Source/target misclassification* is a refined version of untargeted misclassification that specifies which class needs to be predicted for each true class. *Confidence reduction* is the aim to reduce the score representing the confidence of the model in the prediction. Although adjusting

the severity can have some effect on the handling of the identified cases, the score in a pure detection task does not matter within the same predicted class.

2.3 Attack Selection

The attacks in this literature review are selected based on the following criteria: applicability, novelty, and trustworthiness. The applicability ensures that the attacks can be performed on a binary violence image classification model. The novelty requires selected methods to either introduce conceptually new attack classes or to showcase an improvement in a weakness of their predecessor. Finally, the sources of the attacks are filtered to include journal publications or highly cited preprints; however some newer works that have not received much attention yet are also mentioned to outline the current state of research in the field.

The attacks are first split into white box and gray or black box. Then they are grouped by the taxonomy classes introduced in each section or by the underlying concept. Within each group, the attacks are ordered chronologically, demonstrating incremental improvements.

3. White-box Attacks

This section investigates white-box attacks—attacks in which the adversary may use a loss function $L(\theta, \hat{x}, y)$ and the confidence score function f .

Vassilev et al. [15] identify a number of non-mutually exclusive categories of white-box attacks. First, they define *optimization-based* methods, which generate adversarial examples with the objective of minimizing a certain metric of the perturbation, for example, an ℓ_p norm. Then, the taxonomy identifies *universal* evasion attacks, which instead of finding an optimal adversarial example for a specific sample, target the entire data set or a subset with the same perturbation or patch. The next proposed category consists of *physically realizable* attacks which are feasible to implement in the real world, for example, by adding prints and accessories or changing lighting conditions. Finally, the authors mention attacks in *other modalities*. They trick the victim model by presenting malicious data in an unintended form, such as text in image or ASCII art.

Optimization-based attacks pose the highest threat under the condition of full access to the model, since they target imperceptibility without additional constraints. These attacks can be categorized further. For example, a survey by Zhang et al. [16] proposes the following classification based on the type of method used to craft adversarial examples: single-step, iterative, optimization-based, search-based, and generative model-based. However, these classes may overlap and some boundaries are not clearly defined. For example, most optimization-based methods are iterative.

This thesis distinguishes optimization-based attacks based on the nature of constraints and generation strategies. Such categorization results in three main types (Figure 3.1): norm-constrained

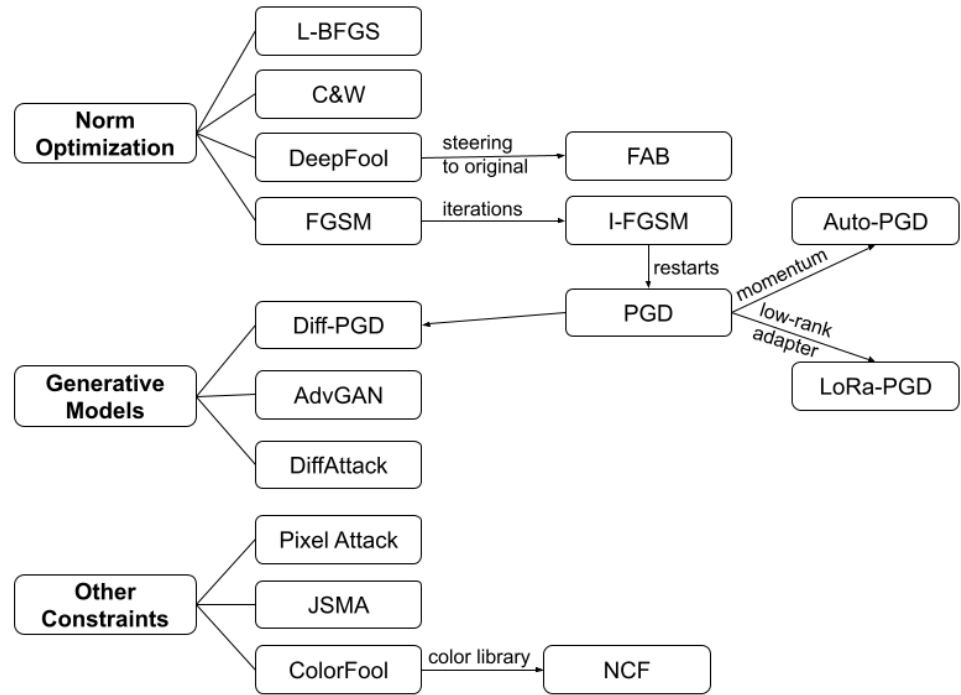


Figure 3.1. Classes, methods, and improvements of white-box attacks

optimization, generative, and attacks under other constraints. Norm-constrained optimization attacks solve an optimization of minimizing a certain norm of the perturbation. Generative attacks utilize generative models trained to produce realistic images according to a learned distribution to find adversarial examples. Attacks under other constraints explore different attack paradigms, such as color distortions and image patches.

3.1 Norm-constrained Optimization

Szegedy et al. [5] defined the attack goal in the following manner:

$$\text{minimize } \|\eta\|_2 \text{ subject to } F(x + \eta) = \hat{y}.$$

Although finding an exact solution is an NP-hard problem for non-trivial classifiers [17], an approximation of the minimum of $\|\eta\|_2^2 + L(\theta, x + \eta, y)$ can be obtained with an optimization algorithm, such as **L-BFGS**.

This algorithm demonstrated high efficiency on the MNIST dataset [18] with AlexNet [1] classifier, successfully attacking all images with an average distortion under 0.1 [5]. However, this approach has some drawbacks. Firstly, the optimization algorithm does not guarantee

that the obtained perturbation is the best globally, as it may find a local minimum. Secondly, the optimization task is computationally expensive as it requires many steps in high-dimensional space, each of which requires model inference.

A powerful tool for constructing adversarial examples is the gradient of the model output with respect to the input image. The **Fast Gradient Sign Method (FGSM)** proposed by Goodfellow et al. [19] utilizes this idea by using the sign of the gradient scaled with a hyperparameter ϵ for the perturbation:

$$\eta = \epsilon \cdot \text{sign}(\nabla_x L(\theta, x, y)).$$

The algorithm is based on the assumption that high-dimensional deep learning models exhibit certain local linearity, and therefore following the gradient is able to alter the confidence scores in a chosen direction. This property allows finding adversarial examples faster than with standard optimization methods, such as L-BFGS, while maintaining a high success rate (over 99.9% with $\epsilon = 0.25$ on MNIST) [19]. Another key difference of FGSM compared to Szegedy et al. [5], is that by taking only a sign of the gradient, FGSM optimizes ℓ_∞ measure of the adversarial perturbation instead of an ℓ_2 norm. This effectively limits the maximum deviation from the original images but affects all pixels and does not consider changes smaller than the selected ϵ .

FGSM is fast, but it does not seek the optimality of the solution. Linear approximations of adversarial perturbations can be refined using an iterative algorithm known as iterative FGSM (**I-FGSM**) [20]. The authors define the adversarial example \hat{x}_t on step t as follows:

$$\hat{x}_0 = x, \hat{x}_{t+1} = P_{B_\epsilon(x)}(\hat{x}_t + \alpha \cdot \text{sign}(\nabla_x(L(\theta, \hat{x}_t, y)))),$$

where P_{x+B_ϵ} is a clip to the ϵ -ball of x .

This approach became the core of the Projected Gradient Descent Attack (**PGD**) [21], which added restarts to prevent the algorithm from getting stuck in local maximums of the loss. On each new start, the authors initialize \hat{x}_0 with a random image within an ϵ -ball of x . This allows the algorithm to explore different paths and select the smallest successful perturbation in the end.

PGD demonstrates higher attack success rate compared to FGSM [21],

particularly on adversarially trained models—models exposed to adversarial examples during the training process. However, PGD is more computationally expensive as it requires many iterations to reach the final solution. Some studies have attempted to improve the efficiency of PGD. For example, **low-rank PGD** attack (LoRa-PGD) [22] was shown to achieve results reaching or surpassing the original algorithm while having a significantly lower memory footprint.

Nonetheless, PGD is not a perfect solution. Despite displaying good performance in regular scenarios, it may fail when the model is adversarially trained against it [23]. Furthermore, PGD requires carefully chosen step size and number of steps to perform optimally. These drawbacks were addressed by **Auto-PGD** [24].

Auto-PGD adds a momentum term to the gradient step [24]:

$$m_{t+1} = P_S(\hat{x}_t + \alpha_t \cdot \nabla L(\theta, \hat{x}_t, y)),$$

$$\hat{x}_{t+1} = P_S(\hat{x}_t + \lambda \cdot (m_{t+1} - \hat{x}_t) + (1 - \lambda) \cdot (\hat{x}_t - \hat{x}_{t-1})),$$

where α_t is the step size at the iteration t , and β controls how much the previous steps affect the current one. The step size is reduced by half at certain moments depending on the optimization progress, and the algorithm is restarted from the best known point.

Different variations of Auto-PGD along with two other methods are combined into a set of attacks called AutoAttack (AA) [24].

Another family of iterative algorithms relying on linearization is based on **DeepFool** [25]. Unlike maximum-confidence algorithms, such as FGSM, PGD, and L-BFGS, DeepFool is a minimum-norm algorithm. This means that it aims to find a minimum perturbation that causes mispredictions rather than attempting to find one under the given norm threshold. This provides a more efficient method to test model robustness as the algorithm does not have to be restarted for each threshold.

The core concept of DeepFool lies in finding a minimal adversarial perturbation for a linear binary classifier, which is a projection on the decision boundary (Figure 3.2) [25]. On each step, DeepFool considers the first order approximation of \mathcal{Z} around \hat{x}_t and computes

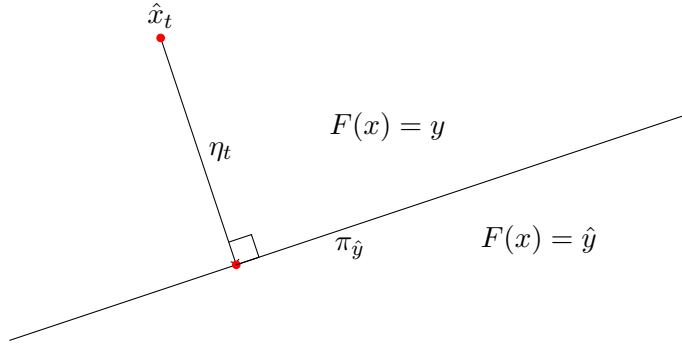


Figure 3.2. Finding a minimum adversarial perturbation for a linear classifier

a minimal perturbation as a solution to

$$\arg \min_{\eta_t} \|\eta_t\|_2 \text{ subject to } \mathcal{Z}(\hat{x}_t) + \nabla \mathcal{Z}(\hat{x}_t)^T \eta_t = 0.$$

The authors derive the solution from the linear case:

$$\eta_t = -\frac{\mathcal{Z}(\hat{x}_t)}{\|\nabla \mathcal{Z}(\hat{x}_t)\|_2^2} \nabla \mathcal{Z}(\hat{x}_t).$$

In the end, the perturbations on each iteration are aggregated into $\hat{\eta} = \sum_t \eta_t$.

DeepFool is fast to compute and effective in generating small adversarial perturbations when the classifier can be well approximated by linear functions. It can also find solutions in highly non-linear cases. However, in those cases, the size of the perturbation may significantly exceed the optimal, as DeepFool minimizes each step rather than the total norm of the perturbation.

Fast Adaptive Boundary Attack (FAB) [17] addresses this shortcoming by introducing an additional incentive for the algorithm to create smaller perturbations. The core algorithm utilizes a projection on the decision hyperplane similarly to DeepFool. However, each step considers both the current point and the original one:

$$\begin{aligned} \hat{\eta}_t &= \text{proj}(\hat{x}_t, \pi_{\hat{y}}, \mathcal{S}) - \hat{x}_t \\ \eta_t^{\text{orig}} &= \text{proj}(x, \pi_{\hat{y}}, \mathcal{S}) - x, \end{aligned}$$

where $\text{proj}(x, \pi_y, \mathcal{S})$ is a projection of x on the decision boundary of the target class π_y within the space \mathcal{S} . The final step is defined by the following formula:

$$\hat{x}_{t+1} = P_{\mathcal{S}} \left((1 - \lambda)(\hat{x}_t + \beta \hat{\eta}_t) + \lambda(x + \beta \eta_t^{\text{orig}}) \right).$$

Additionally, FAB introduces a backward step, which steers \hat{x}_{i+1} closer to the original input if the adversarial example already causes misclassification [17]:

$$\hat{x}_{t+1} \leftarrow (1 - \beta)x + \beta\hat{x}_{t+1}.$$

These modifications reduce the norm of the adversarial perturbation at the cost of slightly slower converging iterations.

Another successor of DeepFool is **SuperDeepFool (SDF)** [26]. It generalizes to various norms and provides an extra projection step to steer the adversarial example to the original image. The authors show that SDF achieves a balance of speed and perturbation size, surpassing most DeepFool-based algorithms in number of iterations and keeping on par with regards to the perturbation norm.

Another prominent attack was introduced by Carlini and Wagner (**C&W**) [27]. The authors define the problem with an objective function ω such that $F(x + \eta) = y \iff \omega(x + \eta) \leq 0$ and a constant $c > 0$:

$$\begin{aligned} &\text{minimize } \|\eta\|_p + c \cdot L(\theta, x + \eta, y), \\ &\text{such that } x + \eta \in [0, 1]^n. \end{aligned}$$

The constant is then chosen to be the smallest value for which $F(x + \eta) = y$. By proposing methods for clipping of $x + \eta$, the authors were able to use Adam optimizer to solve the problem.

This approach demonstrates both good generalization, supporting all of ℓ_0 , ℓ_2 and ℓ_∞ norms, and efficiency, surpassing the success rate of FGSM, DeepFool and JSMA with lower average perturbation norm [27]. Furthermore, the authors show that C&W attack can break defensive distillation, known as a promising defense strategy. However, these results come at a high computational cost of running the optimizer.

3.2 Generative Models

While norm provides an understandable and easily computable metric to measure, it may not be an objective measure of human perception. Thus, certain attacks do not aim to minimize a pure norm of

the adversarial perturbation or any other easily quantifiable metric of the adversarial example. Instead, their goal is to preserve some learned perception of similarity to the original images.

One such method is **AdvGAN** [28] based on generative adversarial networks (GANs). During the training of the generator G and discriminator D , the authors introduce a loss \mathcal{L} consisting of an adversarial loss \mathcal{L}_{adv} , traditional GAN loss \mathcal{L}_{GAN} , and a hinge component $\mathcal{L}_{\text{hinge}}$ on the perturbation size:

$$\mathcal{L}_{\text{GAN}} = \mathbb{E}_x \log D(x) + \mathbb{E}_x \log(1 - D(x + G(x)))$$

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_x L(\theta, x + G(x), y)$$

$$\mathcal{L}_{\text{hinge}} = \mathbb{E}_x \max(0, \|G(x)\|_2 - \epsilon)$$

$$\mathcal{L} = \mathcal{L}_{\text{adv}} + \alpha \mathcal{L}_{\text{GAN}} + \beta \mathcal{L}_{\text{hinge}}.$$

In addition to targeting perceptual stealthiness of the adversarial example by performing evaluation on the discriminator instead of a simple distance measure, AdvGAN provides certain benefits based on the model training. First, once the model is trained, the attack requires only one inference step, which leads to a higher speed. Second, AdvGAN does not rely on the victim model during inference, which lowers the constraints on the model accessibility. However, these benefits come with a higher complexity of the algorithm and the necessity of generation models.

Diffusion models represent another type of generative models suitable for crafting adversarial examples. An example of such an attack is the Diffusion-Based Projected Gradient Descent (**Diff-PGD**) [29], which combines the traditional PGD [21] approach with steps of a diffusion pipeline. The method optimizes the loss on the purified image \hat{x}_t^0 instead of the original \hat{x}_t . The authors perform the purification by applying Stochastic Differential Editing (SDEdit) [30] with K steps on each iteration, so that

$$\hat{x}_t^0 = \text{SDEdit}(\hat{x}_t, K).$$

Then, they perform a regular PGD step:

$$\hat{x}_{t+1} = P_S(\hat{x}_t + \alpha \text{sign} \nabla_x L(\theta, \hat{x}_t^0, y)).$$

Compared to norm-constrained optimization algorithms, Diff-PGD

often demonstrates better stealthiness by generating images without perceivable noise, which is reduced during the diffusion process. Additionally, it is robust to diffusion-based purification, which is an efficient defense method against various types of attacks [31].

Generative-model based attacks do not have to operate on pixel-level intensities. Instead, they can rely on the perturbations in latent space, based on the assumption that similar vectors represent similar content. This idea works well with diffusion models. Chen et al. [32] show that their approach **DiffAttack** of modifying the adversarial examples in the latent space offers an impressive transferability in addition to a high success rate, which makes it a suitable choice for a black-box attack.

The mechanics of DiffAttack [32] is based on inverting the diffusion process to obtain the latent or the noise generating the image. The latent x_t is then perturbed using an optimizer, such as AdamW [33] used in the original experiment, to estimate a latent maximizing the loss of the adversarial example.

3.3 Other Constraints

There are ways to constrain the attacks. For example, one could restrict the number of modified pixels. Although such setting becomes a norm-optimization task with ℓ_0 metric, and can be solved by some norm-optimization methods, such as C&W attack, the non-differentiability of the metric requires special considerations. Thus, ℓ_0 attacks are separated in this section.

Papernot et al. addressed the challenge with the Jacobian saliency map attack (**JSMA**) [34]. As described by the name, they first compute a saliency map of the model predictions $\nabla_x f(x)$. Then, the algorithm identifies the most important pixel—the pixel corresponding to the highest increase in the model confidence score of the target class—and modifies it by a small parameter δ . The process is repeated until the classifier changes the prediction or a limit on the number of modified pixels is reached.

This approach provides a high success rate with only a few pixels altered. However, those changes may be high, which would make the pixels stand out, and the image recognizable as adversarial. This drawback can be addressed with a trade-off on the success rate. Su

et al. demonstrated that a successful attack may require changing only one pixel [35]. Their approach, **One Pixel Attack**, is based on differential evolution (DE).

DE algorithms [36] consider a population (set) of current solutions (parents) and a population of candidate solutions (children). On each iteration, children are compared with their parents and kept only if they demonstrate an improvement on a chosen metric. This approach helps to tackle local minima by maintaining a diverse population and requires less information from the target system as it does not directly rely on model gradients.

One Pixel Attack [35] utilizes DE by encoding the perturbation with the coordinates and RGB intensity values of the modified pixel. The authors use a population of 400 such perturbation candidates and perform up to 100 iterations. According to the study, this approach yields up to 70% success rate on various models evaluated on CIFAR-10 [37]. Furthermore, they extend their approach to multi-pixel edits, resulting in a more general method, Pixel Attack.

Pixel-wise modifications may be noticeable even if the number of modified pixels is low. Moreover, they can be easily avoided by applying denoising or compression algorithms before the inference of the classifier. Therefore, it is useful to consider other constraints, which do not directly translate into the norm of pixel difference. These attacks are called unconstrained.

For example, altering the color distribution of the image is a viable attack direction. However, the color cannot be changed arbitrarily, since certain objects, such as human skin, signs, and food can be linked to a limited set of colors. **ColorFool** [38] addresses this restriction by creating an adversarial coloring with respect to human vision. The authors preserve the pixel lightness by fixing an L-component of the Lab color space [39] and limit the perturbations to a selected natural-looking range based on semantics. The algorithm identifies semantic regions in image with a segmentation model and assigns a color intensity perturbation to each region based on sensitivity. The authors identify four sensitive region types: person, vegetation, water, and sky. Other regions may receive large color intensity changes.

ColorFool yields a high success rate of up to 99% [38] without introducing pixel-wise noise or affecting colors of critical regions.

However, there are some limitations. First, sensitive regions are specified manually, which cannot guarantee that all such regions are properly handled. Second, the perturbations within the regions are chosen randomly [38]. This may result in larger distortions than necessary and affect the perception. A newer method, **Natural Color Fool (NCF)** [40] aims to solve these issues by sampling color distributions from a library based on ADE20K dataset [41] instead of following manually defined ranges. The authors demonstrate that their approach generates higher-quality images, based on NIMA [42] scores, which are designed to estimate human perception of image quality based on deep learning algorithms.

Multiple other unrestricted attacks have been proposed. They rely on features like color and texture [43] or utilize spatial transformations [44].

4. Black and Gray-box Attacks

This section covers black-box and gray-box attacks, for which model weights, confidence scores, and gradients are not available. Gray-box attacks may rely on additional data, such as training sets or model architecture type, while black-box attacks are limited only to the model decision $F(x)$. These attacks can correspond to the following three classes illustrated in Figure 4.1 based on the generation method: transfer-based, processing-based, and query-based. The following subsections cover these classes in detail.

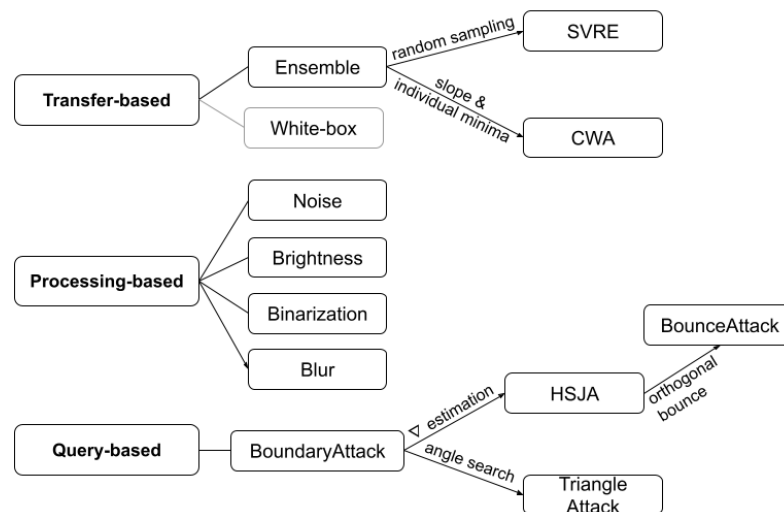


Figure 4.1. Classes, methods, and improvements of black and gray-box attacks

4.1 Transfer-based

Transferability is a property of adversarial examples to work across different models. There are multiple explanations of this phenomenon, such as similar knowledge learned by different models, dense clustering of adversarial examples, and overlap of adversarial subspaces of different models [16]. Transfer-based attacks exploit this property by training separate surrogate models, attacking them, and using obtained adversarial examples to target the victim model.

Transferability was observed in the original discovery of adversarial examples [5]. However, early methods did not put significant focus on improving this property, although some newer methods, for example, generative-model based ones [32] demonstrate high transferability and therefore can be utilized in black-box scenarios.

Attacks that rely on a single surrogate model may be heavily affected by the architecture choices and random factors. These weaknesses may be partially compensated by relying on multiple surrogate models. Dong et al. showed that attacking an ensemble of models led to a higher success rate than generating adversarial examples for one specific surrogate [45]. To perform such an attack, they fused the logits of the models in the ensemble with weights $w_i > 0$, such that $\sum_i w_i = 1$, into

$$\bar{z} = \sum_i w_i z_i$$

and then utilized an optimization-based algorithms, such as FGSM [19], I-FGSM [20], and the newly proposed MI-FGSM. Alternatively, the authors proposed fusing the losses of each model into

$$\bar{L}(\theta, x, y) = \sum_i w_i L_i(\theta_i, x, y),$$

which results in the same attacks as fusing the gradients of the models. Both fusing methods were shown to noticeably outperform the same methods based on a single surrogate.

The main limitation of a simple fusion of logits is that it does not consider the response of each individual model in the ensemble. In ensembles with high model diversity, this may lead to meaningless gradients or generation of images that are not adversarial to the individual models and therefore to a reduction of the success

rate. The former problem was addressed with the **Stochastic Variance Reduced Ensemble Attack (SVRE)** [46], which samples the gradients on each iteration instead of directly fusing all of them. The latter problem was tackled by the **Common Weakness Attack (CWA)** [47]. Its authors propose to use an update step that minimizes distances to the local optimum for each model:

$$\hat{x}_t^i = P_S (\hat{x}_t^{i-1} - \beta \cdot \nabla L(\theta_i, \hat{x}_t^{i-1}, y)).$$

One of the main drawbacks of transfer-based attacks is the complexity of training surrogate models. It requires a dataset similar to the one used during the training of the victim model and may consume a significant amount of computational resources. Furthermore, the success of the attack may depend on the architecture of the surrogate models matching the architecture of the victim model.

4.2 Processing-based

In some cases, normally-occurring image modifications may serve as a basis for evasion attacks. Applying noise, converting to grayscale, changing brightness, and adjusting contrast may cause errors in the model predictions [9]. The magnitude of the changes can be adjusted linearly or with a search algorithm, such as binary search, to find the minimum magnitude leading to a sufficiently high success rate.

The advantage of processing-based methods is the simplicity and accessibility. Most of these transformations are available in image processing libraries, visual editors, or even physical environments. The wide adoption of these transformation may also improve the perception of the images as natural. Thus, the perturbation of many processing-based methods may be imperceptible despite a large number of changed pixels and a high magnitude of change.

Li et al. [9] propose an improvement that may be useful for various types of attacks but in particular for processing-based methods. The improvement is that the attack is performed only on the area that is critical for the decision, which in the case of violence detection, can contain the participants of the fight or of another violent action. The authors argue that these modifications to this area affect the model.

However, this change increases the complexity of the method by introducing additional detection or segmentation models.

The issue with using purely processing-based methods is that the content of the image may become unrecognizable. This changes the true class of the sample, which makes the attack unsuccessful.

4.3 Query-based

Some black or gray-box attacks do not require training of surrogate models. Instead, they query the victim model directly to find patterns in the responses. These methods are referred to as query-based or decision-based.

The first successful decision-based attack, **BoundaryAttack**, was introduced in 2017 [48]. The authors initialize \hat{x}_0 by sampling from a uniform distribution $\mathcal{U}(0, 1)^{H \times W \times C}$ until a sample that belongs to a different class is found. Then, on each iteration of the algorithm they sample η_t from $\eta_t^i \sim \mathcal{N}(0, 1)$ and scale and clip it to follow the conditions

$$\hat{x}_{t-1} + \eta_t \in \mathcal{S} \quad (4.1)$$

$$\|\eta_t\| = \alpha \cdot \|x - \hat{x}_{t-1}\|_2, \quad (4.2)$$

where α is a relative perturbation size. Then, while maintaining the constraint 4.1, they project η_t onto a sphere around original image x and shift \hat{x}_t in the direction of x , so that

$$\|x - (\hat{x}_{t-1} + \eta_t)\|_2^2 - \|x - \hat{x}_{t-1}\|_2^2 = \epsilon \cdot \|x - \hat{x}_{t-1}\|_2^2,$$

where ϵ is a target step towards the original sample.

BoundaryAttack provides a general method for finding adversarial examples, which can achieve perturbation sizes comparable to white-box methods [48], while being model-independent. However, a large number of queries may be required to find optimal adversarial examples, which may not be available in real-world scenarios. The query frequency may be restricted by the performance of the device or a rate limit of a service.

Chen et al. address the high number of queries in **HopSkipJumpAttack (HSJA)** [49] using gradient-direction estimation. On each it-

eration, the method performs a binary search to find the closest point to the decision boundary in the direction of original x , and then samples from a distribution of unit vectors, moves the adversarial example in those directions, and queries the model to get a Monte Carlo estimate of the gradient direction. Consequently, the direction defines where the adversarial example is shifted. The authors demonstrate that HSJA requires significantly fewer queries to achieve adversarial perturbation norms similar to BoundaryAttack.

The HSJA approach can be improved. **BounceAttack** [50] utilizes a gradient-estimation method from HSJA. However instead of using the gradient direction directly, the authors consider only the component orthogonal to the line connecting an adversarial and the original samples. This change is designed to more actively explore the adversarial space, which results in a decrease of the number of performed queries of up to 76% in the experiments.

However, other methods attempt to further reduce the number of queries. One of them, **Triangle Attack (TA)** [51], is based on a geometric idea. The authors argue that in the previous algorithms, BoundaryAttack and HSJA, the angles α_t and β_t between adversarial examples \hat{x}_t and \hat{x}_{t+1} and the original image x satisfy $\beta_t + 2\alpha_t > \pi$ if the new \hat{x} is closer to the original image. Thus, they propose to iteratively sample a subspace and optimize the angles. Their experiments show that Triangle Attack surpasses other state-of-the-art decision-based attacks in the success rate with the same number of queries.

5. Experiment

This section describes an experiment for demonstrating the performance of a subset of the previously identified attacks on violence classification models. The section is split into subsections that describe each step of the experiment. First, the dataset, models, and attacks are defined. The models are trained on selected data and their clean performance is measured. Then, the metrics for evaluating the performance of the attacks are chosen. Finally, the attacks are performed and the metrics are collected and analyzed.

5.1 Methodology

The first step of the experiment is obtaining the violence classification models. In order to ensure their applicability to on-device inference, their size was limited to fewer than 10 million parameters. Under these requirements, the two best performing models according to a survey on lightweight image recognition models [52] were selected: Swiftformer-S [53] and RepViT-M1.1 [54].

The models were originally pre-trained by their authors on ImageNet [55] for general object classification and fine-tuned for the purpose of this experiment on the Real Life Violence and Non-Violence Data [56], which consists of more than 5000 images belonging to each of the positive (violent) and negative (non-violent) classes. The recall scores of the fine-tuned Swiftformer-S and RepViT-M1.1 on the original images are 98% and 95% respectively.

For the selection of attacks included in the experiment, the properties of methods discussed in the literature review are summarized in Table 5.1. Based on these properties, the attacks were filtered in multiple stages. First, the experiment was conducted only on

attacks that do not require additional models because their use is associated with high resource utilization due to the computation cost of training surrogate and generative models or inference of larger models. From the remaining attacks, the methods were selected to include at least one attack from each group present in Table 5.1. In the selection, most recent attacks that support ℓ_∞ norm were prioritized to facilitate a fair comparison of the results; however, the final choice was limited to the implementations available in the Adversarial Robustness Toolbox (ART) [57], which provides a uniform way of performing attacks. As a result, the considered attacks are the following: FGSM as a baseline, Auto-PGD, DeepFool, HSJA, Pixel Attack, and JSMA.

5.2 Metrics

The quality of evasion attacks can be evaluated from different perspectives. This experiment includes metrics from several of them.

- **Stealthiness** or imperceptibility measures how difficult it is to detect the attack. ℓ_p norms of the perturbation can be used as quantifiers; however, another way to measure stealthiness is using an image similarity metric. A structural similarity index measure (SSIM) [58] is a popular option for this task. The main drawback of this metric is that it is patch-based and therefore represents similarity of parts of the image rather than the full perception. Certain metrics, such as the learned perceptual image patch similarity (LPIPS), provide more comprehensive scores by utilizing embeddings of deep learning models. While these models may be vulnerable to adversarial perturbations [59], they solve a different task compared to the victim model and they are not attacked directly. Thus, LPIPS may still provide meaningful stealthiness measurements.
- **Efficacy** is the ability to achieve the target result. In case of violence detection, where the goal of the attacker is to cause misclassification of positive images, Attack Success Rate (ASR) or False Negative Rate (FNR) can be used interchangeably.

$$\text{ASR} = \text{FNR} = \frac{\text{Number of successful attacks}}{\text{Total number of adversarial samples}}$$

Table 5.1. Properties of evasion attacks

Attack	Iterative	Norms	Extra Models	Weights	Confidence ^f	Minimum-norm
L-BFGS [5]	✗	l_2	✗	✓	●	✗
FGSM [19]	✗	l_∞	✗	✓	●	✗
I-FGSM [20]	✓	l_∞	✗	✓	●	✗
PGD [21]	✓	l_∞	✗	✓	●	✗
Auto-PGD [24]	✓	l_∞	✗	✓	●	✗
DeepFool [25]	✓	l_2	✗	✓	●	✓
FAB [17]	✓	l_2	✗	✓	●	✓
SDF [26]	✓	l_2	✗	✓	●	✓
C&W [27]	✓	l_0, l_2, l_∞	✗	✓	●	✓
AdvGAN [28]	✓	l_2, l_∞	GAN	✓	●	✗
Diff-PGD [29]	✓	l_∞	diffusion	✓	●	✗
DiffAttack [32]	✓	unrestricted	diffusion	✓	●	✗
JSMA [34]	✓	l_0	✗	✓	●	✓
Pixel Attack [35]	✓	l_0	✗	✓	●	✗
ColorFool [38]	✓	unrestricted	segmentation	✗	●	✗
NCF [40]	✓	unrestricted	segmentation	✗	●	✗
CWA [47]	✓	N/A	surrogate	✗	○	N/A
SVRE [46]	✓	N/A	surrogate	✗	○	N/A
Processing	✓	unrestricted	segmentation*	✗	○	N/A
BoundaryAttack [48]	✓	l_2	✗	✗	○	✓
HSJA [49]	✓	l_2, l_∞	✗	✗	○	✓
BounceAttack [50]	✓	l_2, l_∞	✗	✗	○	✓
TA [51]	✓	l_2	✗	✗	○	✓

^f Access to the confidence scores of the victim model: ● – always, ○ – never, ◐ – during extra model training. * Optional.

- **Computational Efficiency** is an estimate of the computer resources needed to perform the attack. Common metrics include processor time and asymptotic complexity.
- **Transferability** can be quantified with a transfer attack success rate, which is calculated only on successful adversarial examples against the surrogate model in this experiment to separate transferability from efficacy. It is worth noting that

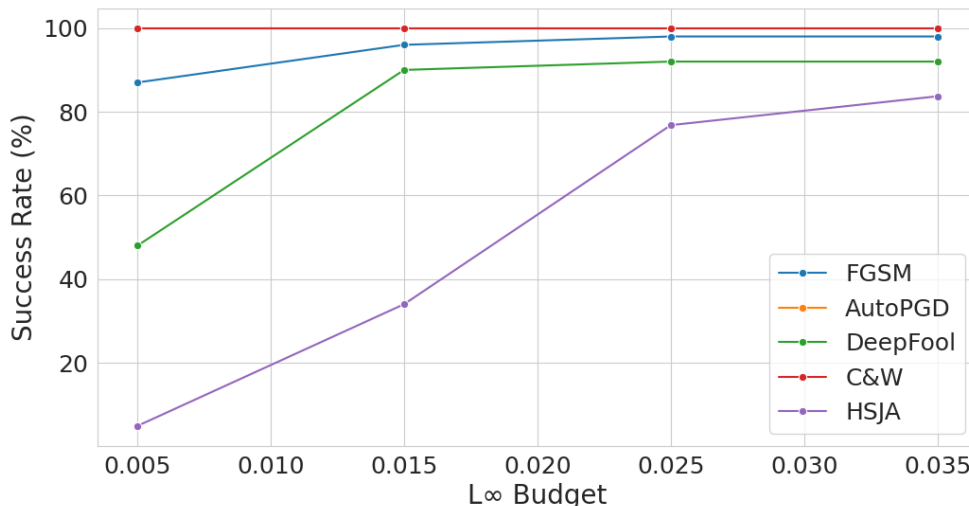


Figure 5.1. Attack success rate on Swiftformer-S depending on the ℓ_∞ budget

this metric may heavily depend on the architecture of both victim and surrogate models.

In order to make a fair comparison, all attacks are given a fixed ℓ_∞ -budget—the maximum allowed ℓ_∞ norm of the perturbation. For fixed budget methods, such as FGSM and Auto-PGD, this constraint is passed directly as a parameter. For minimum-norm methods, the budget is enforced by clipping already generated adversarial examples to an ϵ -ball of the original image.

The attacks were evaluated on a fixed set of 100 randomly selected images from the violent class. The metrics were then aggregated over those images for each attack.

5.3 Results

The attacks that support ℓ_∞ or ℓ_2 norms were first evaluated under different ℓ_∞ -budgets. The success rates by budget are visualized in Figure 5.1. Based on these results, all attacks show a consistent improvement as the budget increases. The changes are the most significant for minimum-norm attacks DeepFool and HSJA; however, C&W attack achieves a stable 100% success rate by finding adversarial perturbations smaller than any of the provided budgets.

For a more detailed comparison, the ℓ_∞ budget was set to 0.015, which allows deviations of pixel intensities of 3 on a 0 to 255 scale while allowing a significant spread of attack success rates based on Figure 5.1. The results of this comparison are presented in the

Table 5.2. Attack results for adversarial examples targeting Swiftformer-S (S) and RepViT-M1.1 (R) defined by column M with the ℓ_∞ budget of 0.015. The stealthiness metrics and time are averaged across the successful adversarial examples. Mean \pm standard deviation are reported for LPIPS.

Attack	M	Success Rate (%)		ℓ_∞	ℓ_2	LPIPS	Time (s)
		Swift.	RepViT				
FGSM [19]	S	87	29	0.015	1.9	0.10 ± 0.06	0.03
	R	89	100	0.015	1.9	0.07 ± 0.05	
Auto-PGD [24]	S	100	78	0.015	3.9	0.03 ± 0.02	4
	R	36	100	0.015	3.8	0.02 ± 0.02	
DeepFool [25]	S	90	0	0.006	0.09	0.00 ± 0.00	0.3
	R	0	88	0.002	0.04	0.00 ± 0.00	
C&W [27]	S	100	4	0.006	1.4	0.01 ± 0.01	130
	R	2	100	0.006	1.5	0.00 ± 0.00	
HSJA [49]	S	34	32	0.015	4.1	0.02 ± 0.02	330
	R	5	77	0.011	2.5	0.01 ± 0.01	

Table 5.3. ℓ_0 attack results for adversarial examples targeting Swiftformer-S (S) and RepViT-M1.1 (R). Mean \pm standard deviation are reported for LPIPS.

Attack	M	Success Rate (%)		ℓ_0	LPIPS	Time (s)
		Swiftformer	RepViT			
JSMA [34]	S	13	0	160	0.02 ± 0.03	5
	R	0	3	120	0.03 ± 0.02	
Pixel Attack [35]	S	39	8	150	0.12 ± 0.10	45
	R	8	14	150	0.12 ± 0.14	

Table 5.2.

Based on the high success rates, both models appear highly vulnerable to most of the attacks. Auto-PGD and C&W are able to generate adversarial examples within the ℓ_∞ budget of 0.015 for all images, while the black-box method HSJA demonstrated noticeably lower efficacy due to stronger constraints.

From the stealthiness perspective, DeepFool provided the least perceptible perturbations based on ℓ_p norms and LPIPS scores, followed by C&W. These attacks also produced the average ℓ_∞ values noticeably below the budget, which indicates that many generated adversarial examples were even smaller than required.

Next, the computational efficiency of the attacks differs significantly. FGSM, being a non-iterative algorithm, executes on average for 30 milliseconds, which is orders of magnitude faster than other attacks, with the closest alternative, DeepFool, performing 10 times slower. However, the inference of the slowest method, HSJA, takes

over 4 orders of magnitude more than FGSM. It is the only black-box attack in this comparison, which extracts less information on each model inference compared to white-box attacks.

While the transferability does not follow a clear pattern based on the model pair, adversarial examples with higher ℓ_2 norms and LPIPS scores tend to exhibit higher transfer success rate. One possible explanation to this phenomenon is that these examples fall out of the distribution of the training data, which was similar for both models.

In addition to ℓ_∞ -constrained attacks, the experiment included two ℓ_0 attacks: JSMA and PixelAttack. Their results summarized in Table 5.3 provide multiple insights. First, their success rate is noticeably lower compared to the methods from Table 5.2 with similar or lower LPIPS scores, which potentially indicates a lower efficacy of the selected ℓ_0 attacks with the same stealthiness properties. Second, JSMA is shown to perform significantly faster than the Pixel Attack at the cost of a slight degradation in success rate and higher ℓ_0 norm. Third, both attacks show low transferability on the selected samples, which might be explained by a limited pixel selection affecting the classification that is specific to each model.

Additionally, manual observation of the adversarial examples generated by the selected ℓ_0 attacks, revealed that the examples can often be easily recognized due to specific pixels noticeably differing from their environment as in Figure 5.2. This effect is also supported by high average LPIPS scores of the Pixel Attack, although it is not prominent in the JSMA results.



Figure 5.2. Adversarial examples of ℓ_0 attacks

6. Conclusion

This thesis identified a wide range of evasion attacks against on-device violence detection models. These attacks covered white-box and black or gray-box scenarios. For each of the scenarios, the attacks were further taxonomized into three classes based on the generation principles and constraints related to adversarial examples. Subsequently, the review demonstrated that each class is comprised of a variety of methods with individual trade-offs and meaningful improvements over the past decade.

The experiment revealed that undefended models are extremely vulnerable to evasion attacks even with a low norm budget. Moreover, some attacks managed to achieve substantial success rates without access to the model weights or confidence scores. However, less constrained attacks displayed better efficacy and stealthiness.

The work is subject to a number of limitations. First, the literature review provides an overview of developed attacks with some concrete examples rather than a comprehensive survey of all available methods. This means that certain useful attack concepts may be missing. Second, the scope and the evaluation set of the experiment are limited due to high computation costs of model inference, varying attack goals, and availability of implementations. Finally, this thesis does not consider certain practical aspects of attack inference, such as physical realizability and quantization of the adversarial examples to the integer intensity values from 0 to 255, which could have an effect on the success rates and inference times.

The main direction of further research on the topic of this thesis is the defense strategies. Given the observed instability of undefended models, these defenses may lead to significant changes of attack success rates and their stealthiness. There exist multiple distinct

types of defenses. Some of them modify the training procedure to improve the model robustness, for example by targeting local prediction consistency [60] or by directly incorporating adversarial examples in the training set [61]. Other methods, such as various input transformations [62], feature squeezing [63], and detection with reformation [64], focus on destroying or identifying adversarial examples during model inference.

References

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.
- [2] A. Esteva et al., "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017, ISSN: 1476-4687. DOI: 10.1038/nature21056.
- [3] N. Youssouf, "Traffic sign classification using CNN and detection using faster-RCNN and YOLOV4," *Heliyon*, vol. 8, no. 12, e11792, 2022, ISSN: 2405-8440. DOI: 10.1016/j.heliyon.2022.e11792.
- [4] T. Sipola, J. Alatalo, T. Kokkonen, and M. Rantonen, "Artificial intelligence in the IoT era: A review of edge AI hardware and software," in *2022 31st Conference of Open Innovations Association (FRUCT)*, 2022, pp. 320–331. DOI: 10.23919/FRUCT54823.2022.9770931.
- [5] C. Szegedy et al., *Intriguing properties of neural networks*, 2014. arXiv: 1312.6199 [cs.CV].
- [6] C. Li, H. Wang, W. Yao, and T. Jiang, "Adversarial attacks in computer vision: A survey," *Journal of Membrane Computing*, vol. 6, no. 2, pp. 130–147, 2024, ISSN: 2523-8914. DOI: 10.1007/s41965-024-00142-3.

- [7] X. Liu et al., "Privacy and security issues in deep learning: A survey," *IEEE Access*, vol. 9, pp. 4566–4593, 2021. DOI: 10.1109/ACCESS.2020.3045078.
- [8] S. Khamaiseh, D. Bagagem, A. Al-Alaj, M. Mancino, and H. Alomari, "Adversarial deep learning: A survey on adversarial attacks and defense mechanisms on image classification," *IEEE Access*, vol. PP, pp. 1–1, Jan. 2022. DOI: 10.1109/ACCESS.2022.3208131.
- [9] X. Li et al., "Adversarial Examples versus Cloud-Based Detectors: A Black-Box Empirical Study," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 04, pp. 1933–1949, Jul. 2021, ISSN: 1941-0018. DOI: 10.1109/TDSC.2019.2943467. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/TDSC.2019.2943467>.
- [10] I. J. Goodfellow et al., "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.
- [11] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 6840–6851. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.
- [12] N. Carlini et al., *On evaluating adversarial robustness*, 2019. arXiv: 1902.06705 [cs.LG].
- [13] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017, pp. 506–519.
- [14] D. Oliynyk, R. Mayer, and A. Rauber, "I know what you trained last summer: A survey on stealing machine learning models and defences," *ACM Comput. Surv.*, vol. 55, no. 14s, Jul. 2023, ISSN: 0360-0300. DOI: 10.1145/3595292.
- [15] A. Vassilev, A. Oprea, A. Fordyce, H. Anderson, X. Davies, and M. Hamin, "Adversarial machine learning: A taxonomy and terminology of attacks and mitigations," National Institute

- of Standards and Technology, Gaithersburg, MD, Tech. Rep. NIST AI 100-2e2025, 2025. DOI: 10.6028/NIST.AI.100-2e2025. [Online]. Available: <https://doi.org/10.6028/NIST.AI.100-2e2025>.
- [16] C. Zhang, L. Zhou, X. Xu, J. Wu, and Z. Liu, "Adversarial attacks of vision tasks in the past 10 years: A survey," vol. 58, no. 2, Sep. 2025, ISSN: 0360-0300. DOI: 10.1145/3743126.
- [17] F. Croce and M. Hein, "Minimally distorted adversarial examples with a fast adaptive boundary attack," in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML'20, JMLR.org, 2020.
- [18] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database," *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, vol. 2, 2010.
- [19] I. J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and harnessing adversarial examples*, 2015. arXiv: 1412.6572 [stat.ML].
- [20] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*, Chapman and Hall/CRC, 2018, pp. 99-112.
- [21] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, *Towards deep learning models resistant to adversarial attacks*, 2019. arXiv: 1706.06083 [stat.ML].
- [22] D. Savostianova, E. Zangrando, and F. Tudisco, *Low-rank adversarial PGD attack*, 2024. arXiv: 2410.12607 [cs.LG].
- [23] A. Mustafa, S. Khan, M. Hayat, R. Goecke, J. Shen, and L. Shao, "Adversarial defense by restricting the hidden space of deep neural networks," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3385-3394.
- [24] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *International conference on machine learning*, PMLR, 2020, pp. 2206-2216.
- [25] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574-2582.

- [26] A. Abdollahpoorrostam, M. Abroshan, and S.-M. Moosavi-Dezfooli, "SuperDeepFool: A new fast and accurate minimal adversarial attack," in *Advances in Neural Information Processing Systems*, A. Globerson et al., Eds., vol. 37, Curran Associates, Inc., 2024, pp. 98 537–98 562.
- [27] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE symposium on security and privacy (sp)*, IEEE, 2017, pp. 39–57.
- [28] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," *arXiv preprint arXiv:1801.02610*, 2018.
- [29] H. Xue, A. Araujo, B. Hu, and Y. Chen, "Diffusion-based adversarial sample generation for improved stealthiness and controllability," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36, Curran Associates, Inc., 2023, pp. 2894–2921. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/088463cd3126aef2002ffc69da42ec59-Paper-Conference.pdf.
- [30] C. Meng et al., "Sdedit: Guided image synthesis and editing with stochastic differential equations," *arXiv preprint arXiv:2108.01073*, 2021.
- [31] W. Nie, B. Guo, Y. Huang, C. Xiao, A. Vahdat, and A. Anandkumar, "Diffusion models for adversarial purification," *arXiv preprint arXiv:2205.07460*, 2022.
- [32] J. Chen, H. Chen, K. Chen, Y. Zhang, Z. Zou, and Z. Shi, "Diffusion models for imperceptible and transferable adversarial attack," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [33] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net, 2019. [Online]. Available: <https://openreview.net/forum?id=Bkg6RiCqY7>.
- [34] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial

- settings,” in *2016 IEEE European symposium on security and privacy (EuroS&P)*, IEEE, 2016, pp. 372–387.
- [35] J. Su, D. V. Vargas, and K. Sakurai, “One pixel attack for fooling deep neural networks,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019. DOI: 10.1109/TEVC.2019.2890858.
- [36] R. Storn and K. Price, “Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 23, Jan. 1995.
- [37] A. Krizhevsky, V. Nair, and G. Hinton, *CIFAR-100 and CIFAR-10 (Canadian Institute for Advanced Research)*, 2009. [Online]. Available: <http://www.cs.toronto.edu/~kriz/cifar.html>.
- [38] A. S. Shamsabadi, R. Sanchez-Matilla, and A. Cavallaro, “Colorfool: Semantic adversarial colorization,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1151–1160.
- [39] D. L. Ruderman, T. W. Cronin, and C.-C. Chiao, “Statistics of cone responses to natural images: Implications for visual coding,” *Journal of the Optical Society of America A*, vol. 15, no. 8, pp. 2036–2045, 1998.
- [40] S. Yuan, Q. Zhang, L. Gao, Y. Cheng, and J. Song, “Natural color fool: Towards boosting black-box unrestricted attacks,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 7546–7560, 2022.
- [41] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Scene parsing through ADE20K dataset,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5122–5130. DOI: 10.1109/CVPR.2017.544.
- [42] H. Talebi and P. Milanfar, “Nima: Neural image assessment,” *IEEE transactions on image processing*, vol. 27, no. 8, pp. 3998–4011, 2018.
- [43] A. Bhattad, M. J. Chong, K. Liang, B. Li, and D. A. Forsyth, “Unrestricted adversarial examples via semantic manipulation,” *arXiv preprint arXiv:1904.06347*, 2019.

- [44] C. Xiao, J.-Y. Zhu, B. Li, W. He, M. Liu, and D. Song, *Spatially transformed adversarial examples*, 2018. arXiv: 1801.02612.
- [45] Y. Dong et al., “Boosting adversarial attacks with momentum,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9185–9193.
- [46] Y. Xiong, J. Lin, M. Zhang, J. E. Hopcroft, and K. He, “Stochastic variance reduced ensemble adversarial attack for boosting the adversarial transferability,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 14 963–14 972. DOI: 10.1109/CVPR52688.2022.01456.
- [47] H. Chen, Y. Zhang, Y. Dong, X. Yang, H. Su, and J. Zhu, “Rethinking model ensemble in transfer-based adversarial attacks,” *arXiv preprint arXiv:2303.09105*, 2023.
- [48] W. Brendel, J. Rauber, and M. Bethge, “Decision-based adversarial attacks: Reliable attacks against black-box machine learning models,” *arXiv preprint arXiv:1712.04248*, 2017.
- [49] J. Chen, M. I. Jordan, and M. J. Wainwright, “Hopskipjumpattack: A query-efficient decision-based attack,” in *2020 IEEE symposium on security and privacy (sp)*, IEEE, 2020, pp. 1277–1294.
- [50] J. Wan, J. Fu, L. Wang, and Z. Yang, “Bounceattack: A query-efficient decision-based adversarial attack by bouncing into the wild,” in *2024 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2024, pp. 1270–1286.
- [51] X. Wang et al., “Triangle attack: A query-efficient decision-based adversarial attack,” in *European Conference on Computer Vision*, 2022.
- [52] Z. Zhang et al., *Image recognition with online lightweight vision transformer: A survey*, 2025. arXiv: 2505.03113 [cs.CV].
- [53] A. Shaker, M. Maaz, H. Rasheed, S. Khan, M.-H. Yang, and F. S. Khan, “Swiftformer: Efficient additive attention for transformer-based real-time mobile vision applications,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 17 425–17 436.

- [54] A. Wang, H. Chen, Z. Lin, J. Han, and G. Ding, "RepViT: Revisiting mobile CNN from ViT perspective," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 909–15 920.
- [55] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.
- [56] M. Soliman, M. Kamal, M. Nashed, Y. Mostafa, B. Chawky, and D. Khattab, "Violence Recognition from Videos using Deep Learning Techniques," in *Proceedings of the 9th International Conference on Intelligent Computing and Information Systems (ICICIS'19)*, Cairo, 2019, pp. 79–84.
- [57] M.-I. Nicolae et al., *Adversarial robustness toolbox v1.0.0*, 2019. arXiv: 1807.01069 [cs.LG].
- [58] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. DOI: 10.1109/TIP.2003.819861.
- [59] F. Croce, C. Schlarman, N. D. Singh, and M. Hein, "Adversarially robust clip models can induce better (robust) perceptual metrics," in *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2025, pp. 636–660. DOI: 10.1109/SaTML64287.2025.00041.
- [60] Y. Tsuzuku, I. Sato, and M. Sugiyama, "Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks," *Advances in neural information processing systems*, vol. 31, 2018.
- [61] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, *Towards deep learning models resistant to adversarial attacks*, 2019. arXiv: 1706.06083 [stat.ML].
- [62] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," in *International Conference on Learning Representations*, 2018. [Online]. Available: <https://openreview.net/forum?id=SyJ7C1WCb>.

- [63] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.
- [64] D. Meng and H. Chen, "Magnet: A two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 135–147.