Generating Adversarial Images in Quantized Domains

Benoit Bonnet, Teddy Furon, Patrick Bas

Generating Adversarial Images in Quantized Domains

Benoit Bonnet[+], Teddy Furon[+], Patrick Bas[†]
[+]Univ. Rennes, Inria, CNRS, IRISA, Rennes, France
[†]CNRS, CRIStAL Lab., École Centrale de Lille, UMR 9189, Lille France

***Abstract*—Many adversarial attacks produce floating-point tensors which are no longer adversarial when converted to raster or JPEG images due to rounding. This paper proposes a method dedicated to quantize adversarial perturbations. This "smart" quantization is conveniently implemented as versatile post-processing. It can be used on top of any white-box attack targeting any model. Its principle is tantamount to a constrained optimization problem aiming to minimize the quantization error while keeping the image adversarial after quantization. A Lagrangian formulation is proposed and an appropriate search of the Lagrangian multiplier enables to increase the success rate. We also add a control mechanism of the ℓ_∞-distortion. Our method operates in both spatial and JPEG domains with little complexity. This study shows that forging adversarial *images* is not a hard constraint: our quantization does not introduce any extra distortion. Moreover, adversarial images quantized as JPEG also challenge defenses relying on the robustness of neural networks against JPEG compression.**

I. Introduction

ALEXNET handily won Imagenet challenge in 2012 [1]. This event is considered to be the turning point of Artificial Intelligence in Computer Vision. Since then, new and deeper neural networks relentlessly improve accuracy on complex datasets with many classes. However accurate these DNNs might be, they can still be surprisingly vulnerable to attacks.

A. Adversarial Attacks

The recent field of adversarial attacks explores ways of fooling DNNs since the work [2]. The goal of an attack is to modify an image with little distortion so that its predicted label differs from the ground truth. The perturbation is *a priori* both classifier and image specific. The literature considers three setups:

- **White-Box**: The attacker knows the classification model architecture and parameters. Most attacks use the very core strength of DNNs to fool them: gradient back-propagation. The very first attacks were FGSM [3], IFGSM [4] and DeepFool [5], later on improved by PGD [6], CW [7], or BP [8].
- **Black-Box**: The attacker only queries the model and observes its output. Attacks can not exploit the gradient. They thus either locally estimate it (HopSkipJump [9] or GeoDa [10]) or probe the class frontier such as SurFree [11].
- **Gray-Box**: The attacker has partial knowledge of his/her target, for instance, the classification model is public but some front-end defense mechanisms are secret.

These attacks are associated with two possible goals:

- **Targeted**: The attacker determines which class should the classifier predict over the adversarial sample.
- **Untargeted**: The attacker only needs the final predicted class to differ from the ground-truth.

This article deals with targeted and untargeted attacks in the white-box scenario.

Recent attacks aim at reducing the distortion (usually measured as ℓ_2 or ℓ_∞ norm), increasing the probability of success, and speeding up the process. Even if some learning procedures result in more robust classifiers [6] and some images are harder to attack, recent white-box attacks craft perturbation invisible to the human eye in most cases provided their complexity budget is large enough.

Regardless their complexity, very few attacks consider the specificity of the medium. A raster image is in its digital form a 3-dimensional matrix of integers, such as the PNG image format. JPEG images [12] are coded as integer matrices representing DCT coefficients in different color spaces. To forge an adversarial *image* rather than just a sample encoded in a floating-point tensor, one needs to craft an integral perturbation. Added to the original image, the result must remain within the defined boundaries (*i.e.* $[0, 255]^N$ with N the number of pixels in the spatial domain).

Attacks rarely address this constraint. It is sometimes argued that the attack is performed inside the classifier in the white-box setup and thus is not required to be integral. While debatable, we consider this assumption to be very niche. Ironically every attack still clips their samples within the boundaries of a pre-processed image (*i.e.* [0,255]). The white-box setup means that the attacker can replicate the model in his/her garage to prepare an attack that will later on be deployed against a remote classifier service analyzing integral images.

The first idea that comes to mind is to round pixel-wise the crafted perturbation to the nearest integer. This is not working. Perturbations are so small that they are partially erased (*i.e.* set to zero) by rounding. Table I gives a first insight of this problem. This preliminary experiment is run on 1,000 randomly selected images from ImageNet. The same images are used throughout this article. The studied classifier is EfficientNet-b0. Rounding an optimized attack significantly

increases the accuracy (*i.e.* decreases the success rate of the attack). An alternative is to round after every step of an iterative attack like PGD_2. This requires significantly more distortion at every iteration so that the perturbation is not erased by rounding. This is displayed as PGD_2 *round* in Table I: it succeeds in beating classification but generates 64% more distortion than our quantization. Figure 1 illustrates further these results.

JPEG compression smoothes the image by cancelling high frequencies, especially at low quality factor. This has little effect on the accuracy of the classifier over natural images while adversarial perturbations are very sensitive to it. Even an attack with an increased distortion budget does not easily fool a classifier after a JPEG compression especially at low-quality factor. Table I shows that JPEG compressing images forged by FGSM or PGD does not create adversarial examples.

This is the reason why some works propose JPEG as a defense against adversarial attacks [13], [14]. Backward Pass Differentiable Approximation (BPDA [15]) was developed to beat this defense and can be used to create JPEG adversarial images as well. BPDA approximates the JPEG compression by a differentiable transformation. This approximation is less accurate for low JPEG quality factors. BPDA then fails to attack some images although it generates more distortion than our quantization. Table I shows that our method JPEG quantizes images which remain adversarial almost surely with a distortion comparable to the compression itself (see section V).

B. Contributions

This article proposes a quantization dedicated to adversarial perturbation so that samples can be saved as adversarial images as shown in Fig.2. It is thus a post-process to be used on top of any attack. This method however relies on gradients available in the white-box setup. It is quick in the sense that it typically needs fewer iterations than the attack per se. Our intensive experimental study shows that forging real images, be it in the raster or JPEG format, is not a hard constraint for the attacker when quantization is properly managed. In other words, our quantization adds little to no extra distortion.

This article is the journal version of the conference paper [16] proposing many improvements. First, quantization is now restricted to a desired range of values providing control over ℓ_∞-distortion (see Sect. III). Second, we extend this method to the JPEG domain (see Sect. IV). This proves to be challenging since this compression erases high-frequencies typical of an adversarial perturbation. Finally we also propose *best-effort* mode for multiple attacks (see Sect. II-D). This mode finds the best parameter setting for each original image in order to reveal the intrinsic power of an attack. This allows a fair comparison of the attacks.

Note that generating adversarial contents in the quantized domain can also be used in another context. The Euclidean distortion can for example be replaced by a steganographic cost to increase the undetectability of adversarial perturbation [17]. The attacker may also target a network which is not a classifier, like a regression function evaluating the visual quality [18].

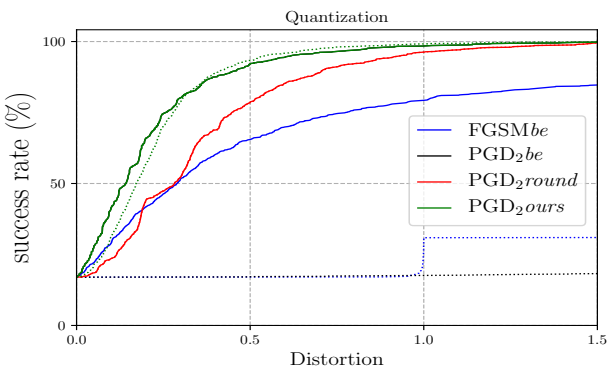

Fig. 1. Operating curves of EfficientNet-b0 against FGSM and PGD in *best-effort mode* with floating point (plain) or quantized (dotted) pixel values.

Fig. 2. Example of adversarial images quantized with our method. Attacked network is EfficientNet-b0. The predicted label is displayed below.

C. Outlines

Section II describes most popular white-box attacks as well as our *best-effort* implementation for each of them. Our evaluation protocol based on operating curves is also explained. Sections III and IV detail our improved post-processing in both spatial and JPEG domains. Finally, section V presents experimental results in various scenarios as well as a thorough study of the impact of each parameter. This further motivates our choice of default parameters. Example images are also displayed throughout this section.

Our code is available at gitlab.inria.fr/bbonnet/adversarial-quantization

II. Attack Model

This section defines notations and attacks used throughout the paper with an emphasis on the concept of *best-effort* mode.

TABLE I
ACCURACY (IN %) AND MEAN AVERAGE DISTORTION OF FGSM AND PGD_2 ATTACKS AGAINST EFFICIENTNET-B0 IN *best-effort* MODE (SEE SECTION II-D) OVER 1,000 RANDOMLY SELECTED IMAGES FROM IMAGENET.

	floats		PNG		JPEG90		JPEG75		JPEG60	
Attack	Acc.	Dist.	Acc.	Dist.	Acc.	Dist.	Acc.	Dist.	Acc.	Dist.
None	83.0	0.00	83.0	0.00	83.0	3.0	81.0	5.2	81.0	6.4
FGSM	11.6	0.33	66.4	0.41	83.0	3.0	81.2	5.2	81.4	6.4
PGD_2	**0.2**	0.16	81.7	0.04	83.0	3.0	81.2	5.2	81.4	6.4
PGD_2 *round*	**0.2**	0.28	**0.2**	0.28	-	-	-	-	-	-
PGD_2 *BPDA*	-	-	-	-	**0.6**	3.9	6.6	5.8	12.5	6.8
PGD_2 *ours*	**0.2**	0.16	**0.2**	0.17	**0.6**	3.0	**0.6**	5.2	**1.0**	6.5

A. Model and Data Notation

Let x_o be an original digital image in the domain $[\![0, 255]\!]^N$ where N is the number of pixels and $[\![0, 255]\!] := \{0, 1, \ldots, 255\}$. This image is preprocessed before feeding a DNN. This stage is defined during the training phase of the DNN to improve its learning capability over the data. After the DNN was successfully trained, it expects data preprocessed in the same fashion to make a prediction. This preprocessing is usually done in two steps: range reduction from $[0, 255]$ to $[0, 1]$, and normalisation:

$$t_o := \frac{x_o/255 - \mu_{data}}{\sigma_{data}}, \tag{1}$$

where μ_{data} and σ_{data} are respectively the mean and standard deviation computed over the training data. These constants are sometimes channel specific (i.e. each channel has its own normalisation), sometimes set to arbitrary values such as 0.5.

Values in tensor t_o are encoded as floating-point variables so that their domain is *pseudo-continuous*. Yet, Equation (1) shows that there are only 256 different possible values for a given entry of t_o. White-box attacks modify t_o into $t_a \in [0, 1]^N$ whose entries may not equal one of the 256 admissible values. This means that by reversing the preprocessing (1), the attacker gets $x_a \in [0, 255]^N$ whose pixel values may not be integers.

For readability, we integrate the preprocessing to our models as the first layer. The sequel focuses on images $x \in [0, 255]^N$ whereas original images are in $[\![0, 255]\!]^N$.

B. Classifier Model

Let $p : [0, 255]^N \to [0, 1]^C$ be a classifier mapping an image to class probability vector p for C classes. The predicted class is defined as:

$$\pi(x_o) := \arg\max_k p_k(x_o). \tag{2}$$

The classifier makes a correct prediction if $\pi(x_o) = c(x_o)$, where $c(x_o)$ denotes the ground-truth class of x_o. An adversarial attack forges an adversarial sample x_a such that:

$$\pi(x_a) \neq c(x_o). \tag{3}$$

The resulting class $\pi(x_a)$ is either chosen by the attacker in a *targeted* scenario or any class that verifies (3) in an *untargeted* scenario. An attack optimizes the perturbation $x_a - x_o$ according to a given metric, usually the ℓ_0, ℓ_2 or ℓ_∞-norm. This gives the following optimization problem on a generic ℓ_m-norm:

$$x_a^\star := \min_{\pi(x_a) \neq c(x_o)} ||x_a - x_o||_m. \tag{4}$$

¡¡¡¡¡¡¡ HEAD In this paper, distortion is measured by the ℓ_2-norm of the perturbation (quantized or not). This is common practice in image processing: The PSNR gives the logarithmic scale of the ℓ_2-norm. Indeed, for natural images (*i.e.* ImageNet, but not MNIST), ℓ_2 reflects distortion perceived by humans *when comparing similar images with very low distortion*. Adversarial images pertain to this low distortion regime.

Attacks in white-box setups are driven by the adversarial loss to craft the perturbation:

$$L_{\text{adv}}(x) := p_{c(x_o)}(x) - p_a(x), \tag{5}$$

where $p_{c(x_o)}(x)$ is the predicted probability of x belonging to the ground truth class $c(x_o)$. The second term $p_a(x)$ is the probability of the adversarial class: In a *targeted* scenario, class a is a parameter of the attack. In an $untargeted$ scenario, it is the best prediction excluding the ground-truth of the original:

$$a = \arg\max_{k \neq c(x_o)} p_k(x). \tag{6}$$

When $L_{\text{adv}}(x) < 0$, the predicted class is not the ground-truth label $c(x_o)$ and x is adversarial. Having access to the model parameters allows to back-propagate differences from the output predictions to the input and to compute the gradient of the loss $\nabla L_{\text{adv}}(x)$.

C. Usual White-box Attacks

Usual white-box attacks use the adversarial loss to point a direction where to look for an adversarial sample and also as a stopping criterion.

1) Fast Gradient Sign Method FGSM: FGSM [3] is the first and most basic attack. It uses the sign of gradient at the original image:

$$x_a = \mathsf{clip}_{[0,255]}\left(x_o - \epsilon \times \mathsf{sign}\left(\nabla L_{\text{adv}}(x_o)\right)\right), \tag{7}$$

where clip_I is the clipping of the component within the interval I. This attack is not iterative, it is a single step process relying on only one parameter ϵ. Note that the adversarial sample is a quantized image only for an integer value of ϵ.

2) Iterative Fast Gradient Sign Method iFGSM: iFGSM [4] is an iterative version of FGSM.

$$x_a^{(i+1)} = \mathsf{clip}_{[0,255]}\left(x_a^{(i)} - \epsilon \times \mathsf{sign}\left(\nabla L_{\text{adv}}(x_a^{(i)})\right)\right). \quad (8)$$

This attack uses two parameters: the descent rate ϵ previously seen in FGSM and the number of iterations N_{iter}.

3) Projected Gradient Descent PGD: This attack is an iterative attack whose updates are defined as follows in its ℓ_2-norm version PGD_2 [6]:

$$x_a^{(i+1)} = \mathsf{clip}_{[0,255]}\left(\mathsf{proj}_\alpha\left(x_a^{(i)} - \epsilon\frac{\nabla L_{\text{adv}}(x_a^{(i)})}{\|\nabla L_{\text{adv}}(x_a^{(i)})\|}\right)\right). \quad (9)$$

PGD_2 revolves around a projection on the ball centered on x_o of radius α. This projection is effective only if the ℓ_2-norm of the perturbation exceeds α. PGD_2 also uses an ℓ_2 normalized gradient to have better control of the perturbation update. This attack uses 3 parameters: the radius α, the descent rate ϵ, and the number of iterations N_{iter}.

4) Boundary Projection BP: BP [8] is a fast two step attack. The first step quickly finds an adversarial sample while the second step refines it by reducing its distortion. Stage 1 is defined as follows:

$$x_a^{(i+1)} = \mathsf{clip}_{[0,255]}\left(x_a^{(i)} - \gamma_i\epsilon\frac{\nabla L_{\text{adv}}(x_a^{(i)})|}{\|\nabla L_{\text{adv}}(x_a^{(i)})\|}\right), \quad (10)$$

where γ_i is an acceleration term ranging from a predetermined γ_{min} at the first iteration to 1 at the final one. Stage 2 refines the adversarial sample found through projection using the same parameters and normalized gradient of L_{adv}.

5) Carlini & Wagner CW: This attack minimizes the following Lagrangian formulation in its ℓ_2-norm version CW_2 [7]:

$$J(x_a, \mu) = ||x_a - x_o||^2 + \mu|L_{\text{adv}}(x_a) - m|_+ \quad (11)$$

where $|z|_+ = \max(0, z)$ and $m \leq 0$ is a margin. The minimum of this equation is found with ADAM optimizer within an inner loop. The outer loop does a line search over μ. When both outer and inner loops are done, the adversarial sample with least distortion is returned. This attack uses five parameters: the numbers of iterations over both loops, the margin m, and the learning rate and momentum for ADAM.

D. Benchmarking Attacks and the "Best-effort Mode"

The last subsection shows that an attack is indeed a family of processes parametrized by one or more parameters. One parameter setting may not be adequate from one classifier to another, and for any image. This explains why experimental results in this literature lack reproducibility.

We propose the concept of *best-effort mode* enabling a fair comparison of attacks and classifiers. It consists of automatically setting the attack to perform as well as possible on a given iteration budget. It finds the parameters such that the attack is successful and the ℓ_2-norm of the perturbation is minimized. (The attack parameters are usually defined within ranges and the optimization may fail providing adversarial).

The implementation of CW is already optimized and therefore needs no tweaking. Two parameters are still defined by the user: boundaries of research and the number of iterations for iterative attacks.

a) FGSM: This attack depends on one parameter ϵ. *Best-effort* simply means running a binary search to find the lowest value that successfully crafts an adversarial perturbation.

b) iFGSM: This attack also depends on one parameter ϵ, for a given number of iterations. We perform the iterative search in the same fashion as previously.

c) PGD: Our *best-effort* mode runs a binary search on the radius α. The iterations budget is equally distributed between the number of iterations and the binary search (e.g. if $N_{\text{iter}} = 100$, 10 radii are tested with $N_{\text{run}} = 10$ iterations each), while ϵ is set to $2\alpha/N_{\text{run}}$. With this value of ϵ, adversarial samples are not projected back onto the ℓ_2-ball of radius α at least within the first half of the N_{run} iterations. Our experiments confirm that this empirical choice is good.

d) BP: This attack leads to very good results when set up correctly. It finds an adversarial sample (stage 1) that is then refined (stage 2). For stability, we aim at finishing stage 1 within roughly the same number of iterations for each image. Inspired by Deepfool [5], this is done through a first-order approximation of the loss:

$$L_{\text{adv}}(x_o + u) \approx L_{\text{adv}}(x_o) + u^\top \nabla L_{\text{adv}}(x_o). \quad (12)$$

Branching this linearisation with (10) gives the value of α cancelling the loss within κ iterations:

$$\alpha = \frac{L_{\text{adv}}(x_o)}{\|\nabla L_{\text{adv}}(x_o)\|_2 \sum_{j=1}^{\kappa}\gamma_j}. \quad (13)$$

We set $\kappa = \lceil 0.2 \times N_{\text{iter}} \rceil$ and experimentally observe that stage 1 is more or less completed when desired, leaving $\approx 0.8 \times N_{\text{iter}}$ iterations to stage 2.

E. Metrics

This paper presents results in two forms: graphs or table. We call *operating curve* the graph showing the success-rate of an attack (y-axis) at a given ℓ_2-distortion (x-axis) such as in Fig. 3. Distortion is measured in the pixel domain $[\![0, 255]\!]$ as

$$d(x_a, x_o) := \|x_a - x_o\|_2/\sqrt{N}, \quad (14)$$

where N is the number of pixels. This is easily interpretable: If for any pixel i, $x_{a,i} = x_{o,i} \pm \epsilon$ (as in FGSM) then $d(x_a, x_o) = \epsilon$. The operating curve sums up the impact of an attack against a classifier over a set of test images $\mathcal{S}_{\text{test}}$ by the following function:

$$d \to P(d) = \frac{|\{x_o \in \mathcal{S}_{\text{test}} : d(x_a, x_o) \leq d\}|}{|\mathcal{S}_{\text{test}}|}. \quad (15)$$

Note that $P(0) = 1 - \eta > 0$, where η is the accuracy of the classifier. This is the fraction of original images which are misclassified, hence considered as already adversarial.

Figure 3 shows operating curves of several attacks against two well-known classifiers. We choose a complexity budget allowing an attack to perform at its best capacity under the best effort mode: FGSM runs on $N_{\text{iter}} = 30$ iterations, BP on

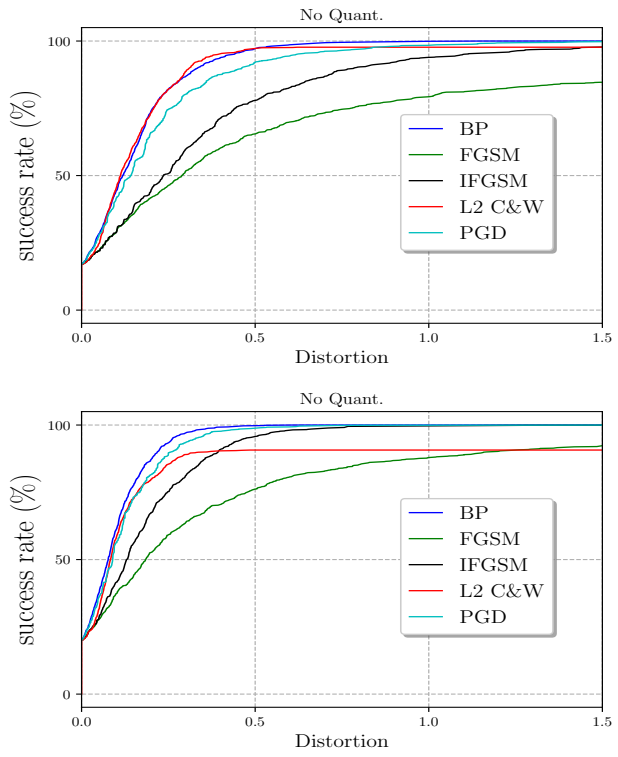

Fig. 3. Operating curves of EfficientNet-b0 (top) and ResNet-50 (bottom) against four attacks in *best-effort mode* and without quantization.

$N_{\text{iter}} = 50$, IFGSM, PGD_2 on $N_{\text{iter}} = 10 \times 10$, and CW_2 on 10×100 iterations (*i.e.* outer loop $\times$ inner loop). We formulate three remarks:

- CW_2 requires even more iterations to successfully attack all images against ResNet-50.
- BP achieves by far the best trade-off between complexity, distortion, and success rate.
- These attacks yield adversarial images with unquantized pixel values, and the average distortion is lower than 0.5 for most images. Rounding these values to $[\![0, 255]\!]$ erases the adversarial perturbation in most pixel positions so that the result is no longer an adversarial image [16].

The next section shows how to modify any white-box attack in order to be robust to quantization in the spatial domain.

III. Smart Quantization in the Spatial Domain

Assume that an attack has forged the adversarial sample $x_a \in [0, 255]^N$, which is not quantized, *i.e.* pixel values are a priori not in $[\![0, 255]\!]$. This section presents our mechanism carefully quantizing the pixel values to keep adversarial images adversarial.

A. Problem Statement

Our mechanism casts $x_a \in [0, 255]^N$ to $x_q \in [\![0, 255]\!]^N$. Its goal is to solve the optimization problem defined in (4) with the additional integral constraint: x_a is eventually replaced by $x_q \in [\![0, 255]\!]^N$. Since we are working in a white-box environment, our method can rely on the following quantities:

- the original image $x_o \in [\![0, 255]\!]^N$,
- the unquantized adversarial image $x_a \in [0, 255]^N$,
- the adversarial loss $L_{\text{adv}}(x)$ and its gradient $\nabla L_{\text{adv}}(x)$.

We introduce the following weak signals:

$$u := x_a - x_o, \tag{16}$$
$$q := x_q - x_a. \tag{17}$$

The quantization noise q plays the central role in our approach. We redefine the distortion and the loss functions w.r.t. this variable:

$$D(q) := \|x_q - x_o\|^2 = \|u + q\|^2 \tag{18}$$
$$L(q) := L_{\text{adv}}(x_q) = L_{\text{adv}}(x_a + q). \tag{19}$$

There is obviously a trade-off between these two quantities. For instance, the choice

$$q^\dagger = \arg\min D(q) = -u \tag{20}$$

cancels the perturbation and makes $x_q = x_o$ not adversarial.

Finding the adversarial image x_q minimizing distortion $D(q)$ can be expressed as:

$$\min_{q \in \mathcal{Q}, L(q) < 0} D(q), \tag{21}$$

where $\mathcal{Q}$ is the set of admissible solutions. Remark that $x_q - x_o = q + u \in \mathbb{Z}^N$ since it is the difference of two integer vectors. This implies that $q \in \mathcal{Q} \subset \mathbb{Z}^N - u$, *i.e.* the grid $\mathbb{Z}^N$ shifted by translation $-u$. For instance, quantizing by rounding the perturbation gives

$$q = [x_a] - x_a = [x_o + u] - (x_o + u) = [u] - u, \tag{22}$$

where $[u]$ is the closest integer value of u component-wise.

We add the other constraint of q being of limited amplitude. We introduce a new parameter $d \in \mathbb{N}_*$, so-called *degree of freedom* which reflects the number of choices per component:

$$q_i + u_i = \begin{cases} [u_i] & \text{if } d = 0 \\ \lfloor u_i \rfloor \text{ or } \lceil u_i \rceil & \text{if } d = 1 \\ [u_i] - 1, [u_i], \text{ or } [u_i] + 1 & \text{if } d = 2 \\ \dots \end{cases}$$

The case $d = 0$ amounts to rounding the perturbation and there is no freedom to choose another integer. This implies that the ℓ_∞-norm of the quantization noise is bounded by $\|q\|_\infty \leq 1/2$. In the general case, this norm is bounded by $\|q\|_\infty \leq (d+1)/2$. The case $d = \infty$ means that the perturbation is quantized to integers but there is no control on the norm $\|q\|_\infty$.

In the end, the set of admissible quantization noises is defined by the product space $\mathcal{Q} = \otimes_{i=1}^N \mathbb{Q}_i$ with

$$\mathbb{Q}_i = \begin{cases} \{[u_i] - \frac{d}{2}, \dots, [u_i] + \frac{d}{2}\} - u_i & \text{if } d \text{ is even,} \\ \{\lfloor u_i \rfloor - \frac{d-1}{2}, \dots, \lceil u_i \rceil + -\frac{d-1}{2}\} - u_i & \text{if } d \text{ is odd.} \end{cases} \tag{23}$$

Note that the number of admissible solutions is exponential with N: $|\mathcal{Q}| = (d+1)^N$.

We now assume that $\forall q \in \mathcal{Q}$, q is small enough to make a first order approximation of the loss:

$$L(q) := L_{\text{adv}}(x_q) \approx L_{\text{adv}}(x_a) + q^\top g, \tag{24}$$

where $g := \nabla L_{\text{adv}}(x_a)$. For instance, the choice

$$q_i^{\ddagger} = \begin{cases} \min(\mathbb{Q}_i) & \text{if } \mathsf{sign}(g_i) > 0 \\ \max(\mathbb{Q}_i) & \text{if } \mathsf{sign}(g_i) < 0 \end{cases} \tag{25}$$

minimizes the first order approximation of $L(q)$. For d large enough, this certainly ensures that $L(q) < 0$ and x_q is adversarial but the distortion is large. The solution to problem (21) can consequently be seen as a compromise between (20) minimizing the distortion and (25) minimizing the loss.

B. Solution

The solution of (21) is given by a Lagrangian formulation. Define the following functional:

$$J_\lambda(q) := D(q) + \lambda L(q), \tag{26}$$

where $\lambda \in \mathbb{R}_+$ is the Lagrangian multiplier balancing adversariality and distortion quantities. Suppose we know how to efficiently minimize that functional by $q_\lambda^\star := \min_{\mathcal{Q}} J_\lambda(q)$, $\forall \lambda \in \mathbb{R}_+$. The expected behavior along λ is for $L(q_\lambda^\star)$ to *decrease* while $D(q_\lambda^\star)$ *increases* (see Fig. 4). For instance,

1) When $\lambda = 0$, all importance is given to $D(q)$. This results in a distortion-based quantization $q^\dagger$ erasing the perturbation u as seen in (20).
2) When $\lambda \to +\infty$, all importance is given to $L(q)$. This results in a gradient-based quantization $q^\ddagger$ of (25).

Since the distortion strictly increases with λ, the optimal solution of problem (21) is then $q_{\lambda^\star}^\star$ where $\lambda^\star = \min\{\lambda : L(q_\lambda^\star) < 0\}$. We practically compute this optimal solution in a two step approach.

1) Minimizing the functional: Finding the minimum of J_λ is difficult, except if we rely on approximation (24), then we can write that

$$J_\lambda(q) \approx \|u + q\|^2 + \lambda L(x_a) + \lambda q^\top g \tag{27}$$

is convex and thus is minimized when $\nabla J_\lambda(q) = 0$. This happens for $q = \tilde{q}_\lambda$, where

$$\tilde{q}_\lambda := -\frac{\lambda}{2} g - u. \tag{28}$$

Yet this solution is not admissible because it does not belong to $\mathcal{Q}$ a priori. We rewrite the approximation (27) as

$$J_\lambda(q) \approx \|q - \tilde{q}_\lambda\|^2 + \frac{\lambda^2}{4}\|g\|^2 + \lambda(g^\top \tilde{q}_\lambda + L(x_a)) \tag{29}$$

to outline that the minimizer on $\mathcal{Q}$ is just its closest element to $\tilde{q}_\lambda$. This amounts to first quantize $\tilde{q}_\lambda$ onto $\mathbb{Z}^N - u$ and then clip:

$$q_{\lambda,i}^\star = \mathsf{clip}_{[\min(\mathbb{Q}_i), \max(\mathbb{Q}_i)]}\left([-\lambda g_i/2] - u_i\right). \tag{30}$$

2) Finding the optimal $\lambda^\star$: The relaxation of the integral constraint and the linearisation of the loss provides a first approximation of $\lambda^\star$. Inserting (28) in (24) yields:

$$\lambda_c := \frac{2(L_{\text{adv}}(x_a) - u^\top g)}{\|g\|^2}. \tag{31}$$

We find the value of λ^* by looking around λ_c. Similarly to our previous work [16], we run a line search in the interval $[0.01\lambda_c, 100\lambda_c]$. For every value tested, we compute the optimal perturbation (30), add it to x_a, and submit this to the classifier. If it is adversarial, then the value of λ is decreased. It is increased otherwise. In other words, the computation of the best quantization noise $q_\lambda^\star$ given λ relies on the linear approximation (24), but the finding of $\lambda^\star$ implies to evaluate the classifier.

IV. Smart Quantization in the JPEG Domain

The JPEG file format represents an image as a 3-dimensional tensor of scaled DCT coefficients quantized to integers. Extending our quantization to the DCT domain enables us to craft JPEG adversarial images.

A. JPEG Compression

The JPEG compression is schematically done in four stages (excluding entropic source coding). A RGB image (Red, Green, Blue) is linearly converted to YC_bC_r (Luminance, blue-Chroma, red-Chroma). This linear transform ensures that all values lie in the range $[0, 255]$. Then each channel undergoes the 8×8 block DCT-transform. The resulting DCT coefficients are finally divided by quantization steps which depend on their frequency bin and the quality factor. A lower quality factor increases the quantization steps s.t. the following quantization loses more information.

This pipeline is linear and thus can be summarized by $X = Jx + C$, where $X \in \mathbb{R}^N$ stores the scaled DCT coefficients and C is a constant vector encoding the shift in the conversion RGB to YC_bCr. We have supposed here that $N = 3LC$ where the numbers of columns C and lines L are multiple of 8. The matrix $J \in \mathbb{R}^{N \times N}$ encodes the color conversion, the block DCT, and the division by the quantization steps. It is cumbersome to express it due to the flattening of images in N dimensional vectors. The main properties are that J is invertible and that it is not an isometry, *i.e.* $\|Jx\| \neq \|x\|$ in general. This is due to the scaling with the quantization steps but also to the color domain conversion [19].

Vectors in this JPEG domain are denoted with capital letters: X_o, X_a, X_q, U, and Q with:

$$U = X_a - X_o = J(x_a - x_o) = Ju, \tag{32}$$
$$Q = X_q - X_a = J(x_q - x_a) = Jq. \tag{33}$$

The scenario is the following: As in the previous section, an attack forges x_a and we have to craft its JPEG version. This amounts to convert it in the JPEG domain, $X_a = Jx_a + C$, and to quantize these coefficients with care so that the image remains adversarial. Note that $X_o = Jx_o + C$ is a priori not an element of $\mathbb{Z}^N$, unless the original image was already quantized in the JPEG domain.

B. Quantization

We write the problem by focusing on the quantization noise Q in the JPEG domain. Since $x_q = J^{-1}(X_q - C)$, Equation (24) is written as:

$$\begin{aligned} L_{\text{adv}}(x_q) &= L_{\text{adv}}(J^{-1}(X_a + Q - C)) = L_{\text{adv}}(x_a + J^{-1}Q) \\ &\approx L_{\text{adv}}(x_a) + (J^{-1}Q)^\top g, \end{aligned} \tag{34}$$

where g is the gradient of the loss function in the pixel domain. As for the Euclidean distortion, we have

$$\|x_q - x_o\|^2 = \|u + q\|^2 = \|u + J^{-1}Q\|^2. \tag{35}$$

Finally, the Lagrangian functional defined in (26) becomes:

$$D(J^{-1}Q) + \lambda L(J^{-1}Q). \tag{36}$$

Following the same reasoning as in the spatial domain, the minimum of this functional when relaxing the quantization constraint amounts to set $J^{-1}Q$ to $\tilde{q}_\lambda$ given in (28). Equivalently:

$$\tilde{Q}_\lambda = -\frac{\lambda}{2} Jg - U. \tag{37}$$

Yet, this time the rounding is done with respect to X_a since we need $X_a + Q$ to be an integral vector:

$$Q^\star_\lambda = \left[-\frac{\lambda}{2} Jg - U + X_a \right] - X_a. \tag{38}$$

Like in the spatial domain, this value is clipped to belong to the set $\mathcal{Q}$, defined in (23) replacing u by U. Note that if the original image is in JPEG format with the same quality factor so that X_o is an integer vector, then $Q^\star_\lambda = \left[\frac{-\lambda}{2} Jg\right] - U$, and we recover a quantization similar to (30).

V. Experimental Work

A. Implementation Details and Setup

We make the following implementation choices.

One can either implement the transformation J^{-1} as a preprocessing layer like previously discussed in Sect. II-A. This allows to directly feed the classifier with JPEG images. Attacks also naturally adapt to this new object as the gradient *back-propagates* through the transformation layer. Yet, this approach makes the attack domain-specific. Our choice of design is to implement our quantization separately on top of any attack. Our method forges JPEG images from a *spatial* adversarial sample x_a resulting from an attack on x_o.

Our implementation builds on the Python library `Pillow` to be as close as possible to the official JPEG standard. Two differences remain: JPEG may apply a sub-sampling on the C_b and C_r channels. Taking into account sub-sampling is straightforward with back-propagation and auto-differentiation. For the sake of simplicity, we work on JPEG images without sub-sampling and the color channels have all the same size. JPEG may apply clipping when converting from one color domain to another. These border effects produce small information losses. We do not apply this lossy step to keep the transformation linear.

The experiments use 1,000 PNG versions of images from the validation set of Imagenet ILSVR 2012 [20]. Unless stated otherwise, the attacked classifier is EfficientNet-b0 [21]. EfficientNet in its b0 configuration is a recent and lightweight classifier that achieves high accuracy on ImageNet. Our previous work [16] shows that distortion created by quantization is proportional to the distortion created by the attack. We thus choose BP to be the default attack. It performs well with few iterations in its *best-effort* setup as seen in Fig. 3. The research on the optimal value of λ is done by default over 10 iterations. Finally, both spatial and JPEG quantization are run by default with degree of freedom $d = 1$ unless specified otherwise.

The protocol is the following. For the spatial domain, $x_q = x_o$ if the original image is already misclassified, otherwise BP produces x_a that our method quantizes to x_q. For the JPEG domain, x_o is first converted in the JPEG format. If this triggers a misclassification, then this JPEG version of x_o is the adversarial image X_q. Otherwise, BP produces x_a from the JPEG original that our method quantizes to X_q.

Operating curves in both domains are displayed as follow:

- Distortion is calculated w.r.t. the original spatial image.
- Misclassified original images in each domain are considered as already adversarial at null distortion.

Note that compressed JPEG images do not have a null distortion since they differ from the original spatial image. For the sake of clarity, we however consider they do. This choice is further motivated in Section V-C.

B. Investigations on the Search of $\lambda^\star$

Figure 4 shows the behavior of the adversarial loss and the distortion as functions of λ, for one image in the spatial and the JPEG domains. To plot these curves we use the quantized (30) (resp. (37) for JPEG) and unquantized solution (38) (resp. (28)). The same behavior is observed on other images through other classifiers up to a change of ranges of values.

Without quantization, the distortion is the same in both domains and it strictly increases w.r.t. λ as predicted by (28) or (37). With quantization and clipping (with $d = 1$), the distortion increases much more in the JPEG domain because of the coarser quantization steps in the high-frequency bins. It also does not start at 0 but at the distortion induced by the regular JPEG compression of the original image.

In the spatial domain, the adversarial losses (with or without quantization and/or clipping) are well-approximated by (24) for small perturbation, *i.e.* when λ is small. In particular, they converge to $L_{\text{adv}}(x_o)$ when $\lambda \to 0$. The linear approximation is useful for predicting when the loss cancels. Adding quantization and clipping constrains the problem and we observe that $\lambda^\star > \lambda_c$. This implies a stronger distortion. Of course, the linear approximation is very wrong when predicting losses below -1.

The picture is less clear in the JPEG domain. The approximation holds true in the beginning and until $L_{\text{adv}}(X_q)$ reaches 0. The approximation λ_c remains extremely relevant. However L_{adv} sometimes becomes non-monotonic as $\lambda \to \infty$ because of the strong distortion.

For this reason, our search of λ^* slightly differs. It starts from λ_c given in (31). This is the same value for both spatial and JPEG domains. In the spatial domain, a line search within

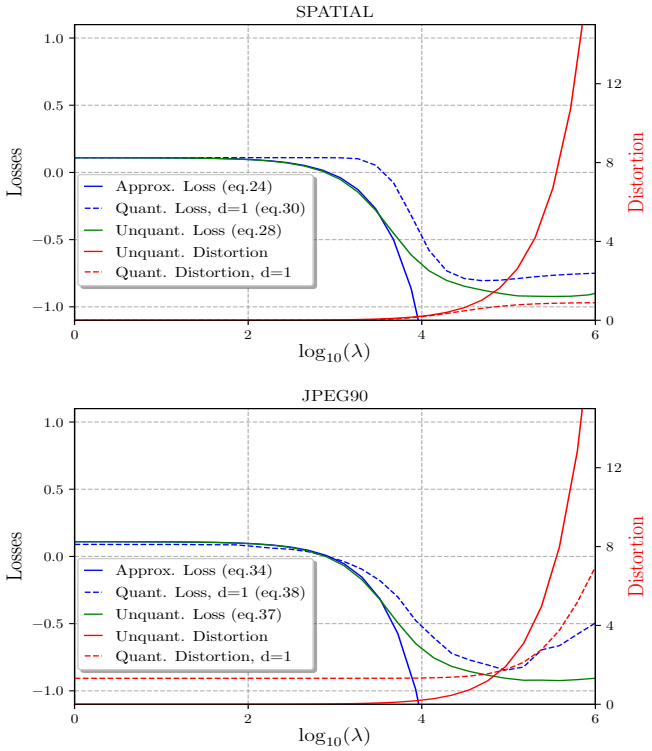

Fig. 4. Comparing the approximated loss (24) (resp. (34)) with the loss without rounding: $L(\tilde{q}_\lambda)$ in (28) (resp. $L(J^{-1}\tilde{Q}_\lambda)$ in (37)) and the loss with quantization $L(q^\star_\lambda)$ in (30) (resp. $L(J^{-1}Q^\star_\lambda)$ in (38)) as a function of λ in the (top) spatial domain, resp. (bottom) JPEG90. Distortion is also displayed with scale on the right.

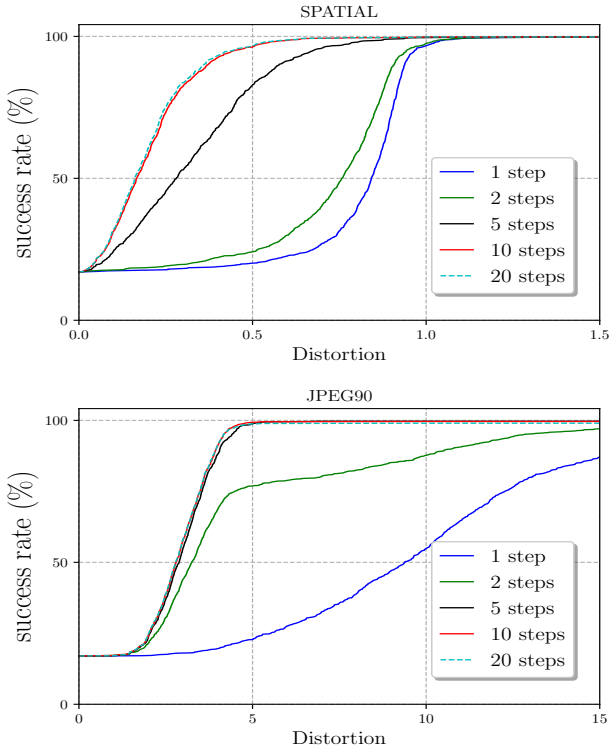

Fig. 5. The impact of the number of tested values of λ on the operating curve. EfficientNet, BP, $d = 1$, spatial (top) and JPEG90 (bottom).

$[0.01 \cdot \lambda_c, 100 \cdot \lambda_c]$ works well because the loss is almost monotonically decreasing. In the JPEG domain, the loss is less predictable and we instead sample n values in this interval:

$$\lambda_i := \lambda_c \cdot 10^{\alpha_i}, \tag{39}$$

$$\alpha_i := 2 \cdot \frac{n-2i}{n}. \tag{40}$$

The lowest value tested that verifies $L_{adv} < 0$ is necessarily the best since distortion strictly increases with λ.

It is expected that the line search in the spatial domain is more efficient than the uniform sampling in the JPEG domain. This is indeed illustrated by Fig. 5. Quantization in the spatial domain converges faster thanks to the line search. However, for both approaches, searching for $\lambda^\star$ with $n = 10$ steps is sufficient. Note that each step only makes a forward pass through the classifier network. In comparison running BP makes $N_{\text{iter}} = 50$ forward and backward passes. Our quantization is thus faster than the attack. It gets slowed down by the DCT transform in JPEG domain however.

C. Impact of Quality Factor in JPEG Domain

The quality factor of JPEG determines the values of the quantization steps applied on the DCT coefficients. A lower quality factor leads to a bigger loss in information and degradation of the image. This is especially true for high frequencies of the image which are coarsely quantized. The gradients computed during attacks look like noisy patterns and the resulting adversarial perturbation is thus in the high-frequency range. Its distortion needs to be amplified to preserve the adversarial property of the perturbation at a lower quality factor.

Figure 7 shows operating curves for adversarial images crafted for different JPEG quality factors (plain curves) which confirm this last statement. For the sake of comparison, the distortion of the JPEG compression on the original images is also displayed as a cumulative sum over all the 1,000 images (dashed curves). We observe that our adversarial quantization returns a distortion very close to regular JPEG compression.

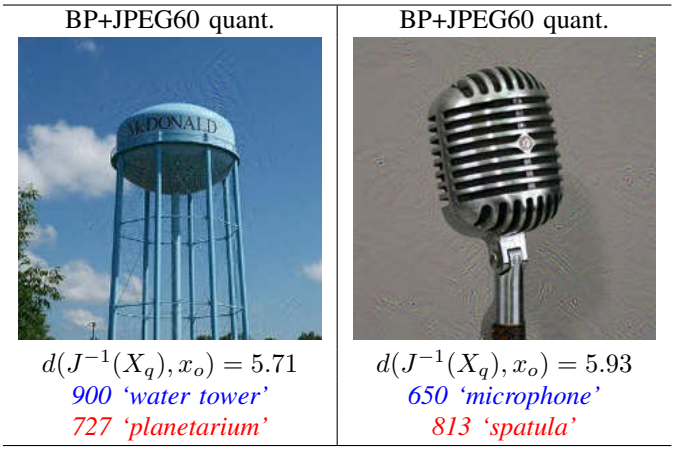

Fig. 6. Visible artifacts on images quantized as JPEG60. Predicted class is displayed in red, ground-truth in blue.

This leads to the interesting result that some adversarial images quantized in JPEG often have the same distortion (or even *less* in few cases) than just the original image compressed. Figure 14 illustrates this result with JPEG75. On three of the four examples displayed, distortion of the adversarial image is equal or very close to the distortion of the compressed image. In other words, a perturbation being compliant to JPEG is not a strong constraint for the attacker as it is almost distortion-free.

The price to pay is a small extra complexity thanks to our method. This is necessary as the JPEG compression alone is not working as an attack. Table **??** indicates that only 17.0% (JPEG90 and JPEG100) or 19.0% (JPEG75 and JPEG60) of compressed original images are misclassified. We plot the same operating curve as previously considering JPEG compression as an attack on Fig. 7 (dash plots). Its discloses which images were naturally adversarial after JPEG compression and which ones needed to be attacked and quantized by our method. There is no correlation between the distortion due to JPEG and the chance that it succeeds as an attack. Indeed, most of these images are already misclassified in PNG and still adversarial once in JPEG.

Figure 8 shows the operating curve w.r.t. to the Euclidean distance from the original image *compressed to JPEG format*. These curves start at a success rate of 17.0% (resp. 19.0%) since a null distortion corresponds to misclassified original JPEG images. Except for these specific images, our method forges an attacked JPEG image different than its original JPEG version although both of them are equivalently far away from the original PNG image.

Distortion is mostly imperceptible when the quality factor is high. It does start to be noticeable with JPEG75. Fig. 14 displays for example at the last row some little artifacts at the bottom of the lighter which are not typical from a JPEG compression. On JPEG60, quantization artifacts are more frequent and important. Figure 6 displays two examples. These patterns are especially visible on smooth image regions usually not affected by JPEG compression. Quantized attacks remain imperceptible when the image is highly textured.

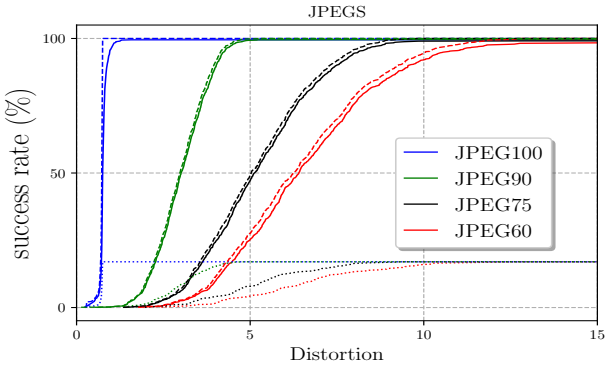

Fig. 7. The operating curve of Efficientnet-b0 against BP + JPEG quantization with $d = 1$ (plain) and against JPEG compression alone (dot). For the sake of comparison, the cumulative distribution function of the distortion due to JPEG compression is also displayed (dashed). Distortion is measured from the original spatial image x_o.

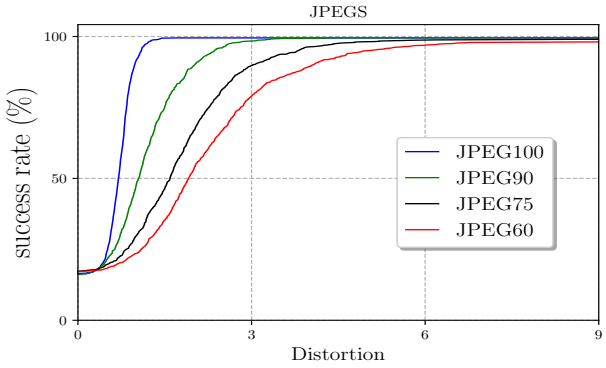

Fig. 8. The operating curve of Efficientnet-b0 against BP + JPEG quantization ($d = 1$). Distortion is calculated w.r.t. to the original image compressed in JPEG domain X_o.

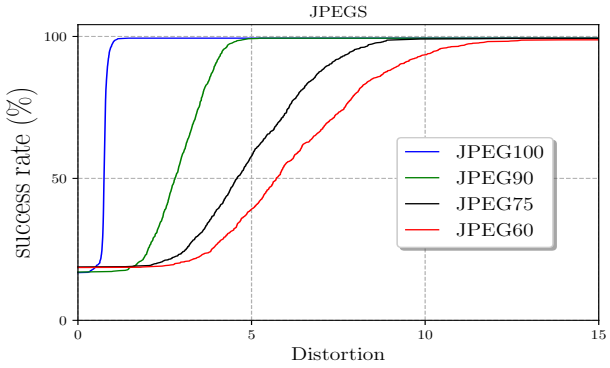

Fig. 9. The operating curve of Efficientnet-b0 against BP + JPEG quantization ($d = 1$). Distortion is measured from the original spatial image x_o. Images that are already adversarial after compression are considered to have null distortion for readability.

We conclude this section by the following remarks.

- The classifier is very robust to JPEG compression alone.
- For any tested quality factor, almost all images are successfully attacked and quantized. A few images were unsuccessful at low quality factors (1% for JPEG60).
- Little distortion is added to the inherent distortion of the JPEG compression.
- Yet, at low quality factor, the adversarial perturbation is not typical from JPEG compression artefact.

Figure 9 summarizes the results but this time assuming that misclassified images have null distortion. We consider this procedure to carry out the main information about the efficiency of the attack and we use this setup to display the following results.

D. Impact of the Degree of Freedom

Our previous work [16] considers only quantization in the spatial domain with $d = 1$. The method here is more general with higher degrees of freedom. Is that useful?

When $d = 0$, quantization is equivalent to rounding the perturbation samples to the nearest integer. Figures 13 shows that this mostly leads to unsuccessful attack. Indeed, the perturbation is a weak signal partially destroyed by rounding.

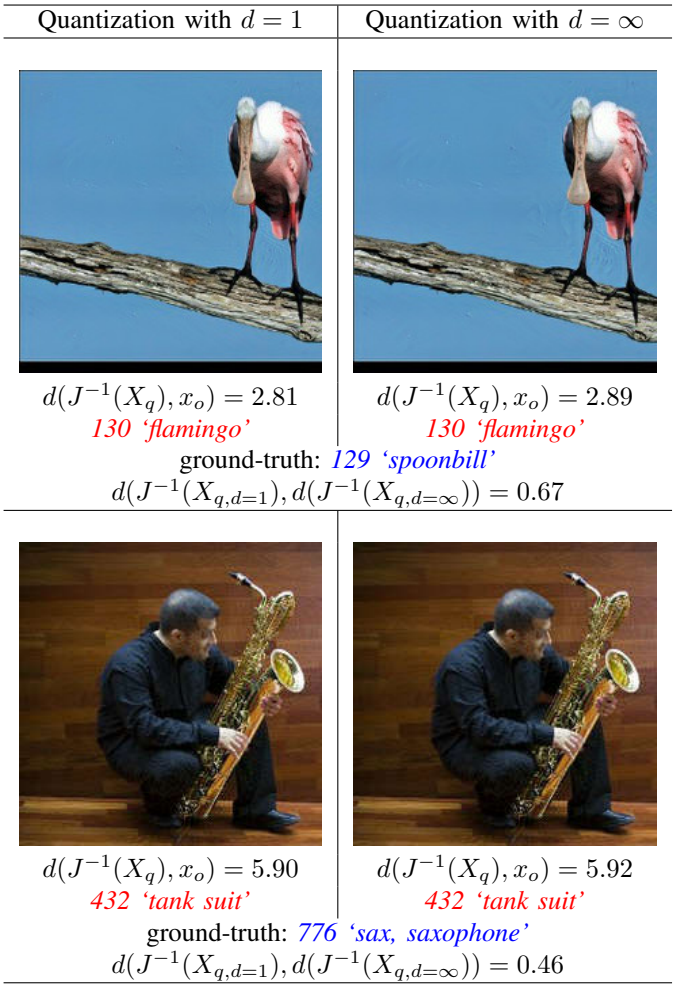

Fig. 10. Visual artefacts for two adversarial images on JPEG90 (top) and JPEG60 (bottom) with and without clipping.

The success rate converges to approximately 20% for the different JPEG quality factors, close to the proportion of misclassified original images (see Fig. 7). This demonstrates the robustness of the classifier against JPEG.

When attacking EfficientNet-b0, $d = 1$ seems enough to quantize almost every image in both domains and the benefit of increasing d seems rather low. A higher value of d is not necessarily a better choice. This is particularly visible in Fig. 13 with the quality factor 100.

The reason lies in the metrics used. The distortion (14) is proportional to the ℓ_2-norm, *i.e.* the square root of the squared-difference, summed over all pixels. Adding $+2$ on one coefficient thus costs 2 whereas adding $+1$ on two coefficients costs $\sqrt{2}$. The degree of freedom constrains the ℓ_∞ norm of the total perturbation $u + q$ (or $U + Q$ in the JPEG domain). This clipping increases the spreading of the perturbation over all the coefficients: since the coefficients with large gradient amplitude can not host a large perturbation, λ increases to compensate this clipping on the other components. This more uniform distribution of the perturbation energy over the coefficients yields a lower Euclidean distortion but also a lower perceptual impact. Figure 10 shows two images quantized with two different quality factors each quantized with $d = 1$ and $d = \infty$. Their Euclidean distortion w.r.t. the original image is similar but the artefact are more visible for $d = \infty$ (quantization but no clipping). We notice that this holds for any image.

E. Quantization on Different Classifiers and Attacks

This study considers four recent classifiers: EfficientNet-b0 [21] and its adversarially trained counterpart EfficientNet-b0-advprop [22]; RegNetX-032 [23], and the older ResNet50 [24].

Figure 12 shows two images misclassified by two different classifiers. It is interesting to note that both neural networks misclassified the right image as the same class (828: tray) whereas the left image is misclassified with different labels yet semantically very close. These classification errors are also understandable from a human point of view. As a final comment on classifiers: all images misclassified by EfficientNet-b0 (170 total) are misclassified by RegNetX-032 as well, and RegNetX-032 misclassified 6 more images (176 total).

Figure 11 shows the operating curves of all four classifiers. They are attacked with BP or PGD_2 and quantized in both spatial and JPEG90 domains. Gradients from one classifier to another vary with the number and nature of hidden layers. This affects how an attack behaves. The hierarchy between unquantized attacks however remains the same from one classifier to another as seen in Fig. 3. This order remains after quantization in the spatial domain: BP outperforms PGD_2.

Nevertheless, the differences between classifiers and attacks are barely noticeable in the JPEG domain. Distortion added by JPEG compression takes over as Sec. V-C explains and imposes the common shape of the operating curve. This shows the adaptability of our quantization w.r.t. which neural network is attacked.

F. Transferability

Our quantization method aims at creating an adversarial image with minimum distortion. The image therefore lies just behind the frontier of correct classification for the targeted classifier. Therefore, no transferability to other deep neural networks is guaranteed. We consider the scenario where the attacker knows that the deployed classifier belongs to an ensemble but she/he does not know which one exactly. The goal is to forge images adversarial for all the classifiers in the ensemble. Our strategy is to aggregate the losses of the classifiers with the maximum operator so that we focus on the most robust element of the ensemble and to aggregate their gradients with the average operator like in Expectation over Transformation (EoT [15], [25]). Figure 11 shows that beating all the classifiers in the ensemble does not amount to beat the most robust one (*i.e.* Efficientb0 advprop). More distortion is required instead in spatial domain. In JPEG domain we observe that the slope of the ensemble curve is similar to the curve of any single model. The distortion created by the quantization is still on par with compression alone. We do note however that quantizing for several classifiers is a more difficult task in JPEG. The final accuracy is 6.7% in JPEG, 0% in spatial domain.

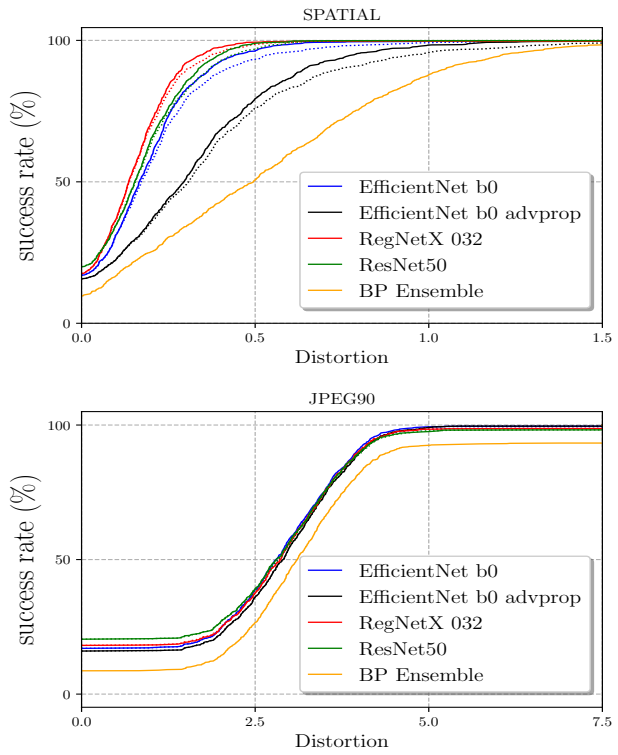

Fig. 11. The operating curves of several classifiers against BP (plain) and PGD_2 (dotted) with quantization in spatial (top) and JPEG90 (bottom) domains.

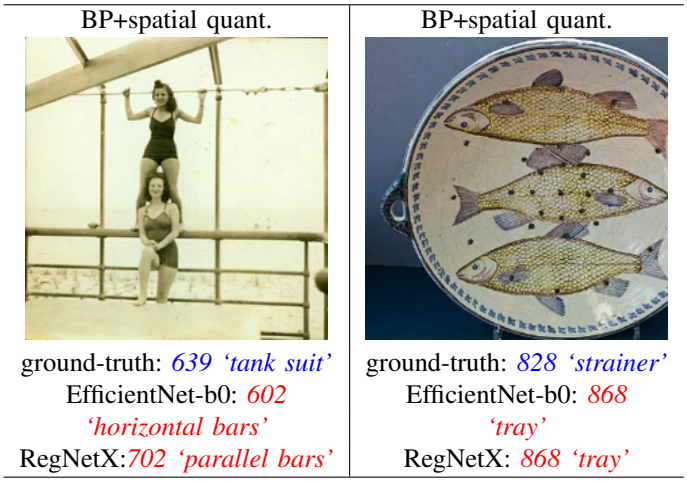

Fig. 12. Misclassification of two images from our dataset through two classifiers: EfficientNet-b0 and RegNetX-032.

G. JPEG Compression as a Defense

As mentioned several times throughout the paper, JPEG compression usually erases adversarial perturbations while not spoiling the accuracy of the classifier over natural images. For this reason, JPEG compression has been studied as a means of defense against adversarial samples [13], [14]. It acts as a low-pass filter reforming the input image. Table II shows this is indeed true for our spatially quantized images. While a quality factor of 90 reforms 42% of our adversarial PNG images, a quality factor of 60 reforms $\approx 70\%$. This proves to be a very effective defense against adversarial samples.

TABLE II
ACCURACY (IN %) OF EFFICIENTNET-B0 EQUIPPED WITH A JPEG COMPRESSION AS A DEFENCE FRONT-END REFORMER AGAINST OUR QUANTIZED *best-effort* BP.

Atttack	Defense None	JPEG100	JPEG90	JPEG75	JPEG60
Spatial	**0.1**	4.0	42.1	63.0	70.4
JPEG100	0.6	1.0	39.8	71.4	76.6
JPEG90	0.6	**0.7**	**0.6**	60.0	69.8
JPEG75	0.6	**0.7**	2.4	**0.6**	14.7
JPEG60	1.0	1.0	1.3	6.4	**1.1**

However, our adversarial images quantized in the JPEG domain are naturally more robust to JPEG compression. The results show interesting properties:

- The performance of the attack is maximized when the quality factor matches the one used at the defense.
- Compressing at the same quality factor does however reform few images ($< 1\%$ in all three cases) because JPEG is not idempotent.
- Quantized adversarials at a given quality factor are robust to defenses with a higher quality factor.

VI. CONCLUSION

We have proposed a method (improved from [16]) to effectively quantize adversarial samples in order to craft adversarial images in spatial or JPEG domains. This quantization guarantees that generated images remain adversarial while minimizing the distortion. It runs within few *forward* calls to the network making it faster than simple attacks and it conveniently operates on top of any white box attacks for broader usability.

When dealing with JPEG compression, the distortion induced by the attack is a very small fraction of the distortion induced by sole compression, and crafting adversarial images in JPEG at low-quality factors also provides robustness to countermeasures based on JPEG compression.

The presented methodology is moreover ubiquitous and it could be transferred to other domains such as JPEG2000 [26] or HEIF [27], and the optimization setup can also be used for other metrics than classification (for example regression as proposed in [18]) and on other distances such as steganographic costs [17] in order to generate adversarial images that are less prone to be statistically detected.

VII. ACKNOWLEDGMENTS

The ph.d. thesis of B. Bonnet is funded thanks to a grant from Direction Générale de l'Armement, French Ministry for the Armed Forces. This work has been funded in part by the French National Research Agency under the ALASKA project (ANR-18-ASTR-0009), the DEFALS program (ANR-16- DEFA-0003), and in part by the French Agence Innovation pour la Défense under the chaire SAIDA (ANR-20-CHIA-0011-01). This work was granted access to the HPC resources of IDRIS under the allocation AD011011402R1 made by GENCI (Grand Equipement National de Calcul Intensif).

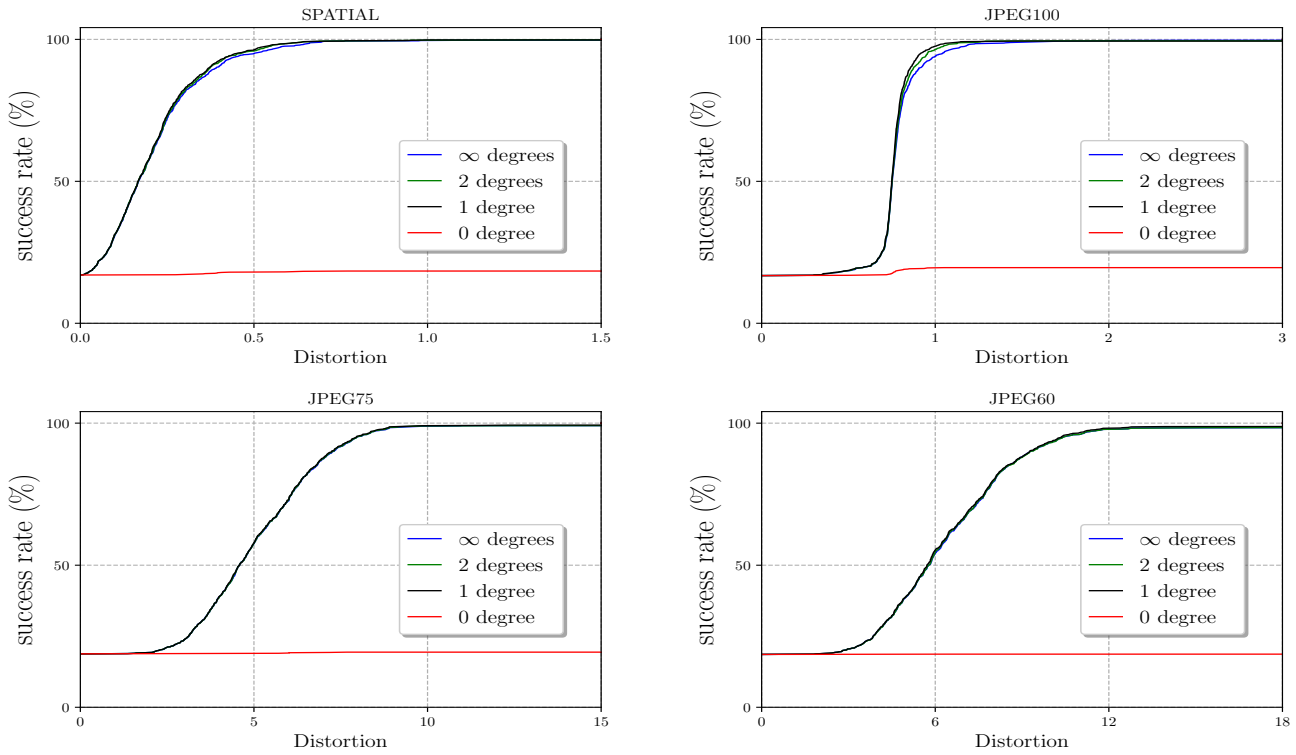

Fig. 13. The operating curves of EfficientNet-b0 against BP in the spatial and JPEG domain with different degree of freedom. Distortion is measured from the original spatial image.

REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[2] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[3] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," in *ICLR 2015, San Diego, CA, USA,*, 2015.

[4] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al., "Adversarial examples in the physical world," 2016.

[5] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal 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.

[6] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR 2018, Vancouver, BC, Canada.*, 2018.

[7] Nicholas Carlini and David Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symp. on Security and Privacy*, 2017.

[8] H. Zhang, Y. Avrithis, T. Furon, and L. Amsaleg, "Walking on the edge: Fast, low-distortion adversarial examples," *IEEE Trans. on Information Forensics and Security*, 2020.

[9] Jianbo Chen, Michael I Jordan, and Martin J Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," in *2020 ieee symposium on security and privacy (sp)*. IEEE, 2020, pp. 1277–1294.

[10] Ali Rahmati, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Huaiyu Dai, "Geoda: a geometric framework for black-box adversarial attacks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8446–8455.

[11] Thibault Maho, Teddy Furon, and Erwan Le Merrer, "Surfree: a fast surrogate-free black-box attack," *arXiv preprint arXiv:2011.12807*, 2020.

[12] Gregory K Wallace, "The jpeg still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[13] Uri Shaham, James Garritano, Yutaro Yamada, Ethan Weinberger, Alex Cloninger, Xiuyuan Cheng, Kelly Stanton, and Yuval Kluger, "Defending against adversarial images using basis functions transformations," 2018.

[14] Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen, "Feature distillation: Dnn-oriented jpeg compression against adversarial examples," 2019.

[15] Anish Athalye, Nicholas Carlini, and David Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International conference on machine learning*. PMLR, 2018, pp. 274–283.

[16] Benoît Bonnet, Teddy Furon, and Patrick Bas, "What if adversarial samples were digital images?," in *Proc. of ACM IH&MMSec '20*, 2020, pp. 55–66.

[17] B. Bonnet, T. Furon, and P. Bas, "Adversarial images through stega glasses," in *submitted to IEEE WIFS'20*, 2020.

[18] Benoit Bonnet, Teddy Furon, and Patrick Bas, "Fooling an Automatic Image Quality Estimator," in *MediaEval 2020 - MediaEval Benchmarking Intiative for Multimedia Evaluation*, Online, United States, Dec. 2020, pp. 1–4.

[19] Shuyuan Y. Zhu, Zhiying Y. He, Chen Chen, Shuaicheng C. Liu, Jiantao Zhou, Yuanfang Guo, and Bing Zeng, "High-quality color image compression by quantization crossing color spaces," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 5, pp. 1474–1487, 2019.

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[21] Mingxing Tan and Quoc V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv*, 2019.

[22] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan Yuille, and Quoc V. Le, "Adversarial examples improve image recognition," *arXiv*, 2019.

[23] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár, "Designing network design spaces," 2020.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image

Natural	BP+spatial quant.	JPEG75 compression	BP+JPEG75 quant.
			
2 'white shark' $d(x_q, x_o) = 0.0$	*3 'tiger shark'* $d(x_q, x_o) = 0.02$	*2 'white shark'* $d(J^{-1}(X_q), x_o) = 3.15$	*3 'tiger shark'* $d(J^{-1}(X_q), x_o) = 3.15$
			
791 'shopping cart' $d(x_q, x_o) = 0.0$	*161 'basset hound'* $d(x_q, x_o) = 0.53$	*791 'shopping cart'* $d(J^{-1}(X_q), x_o) = 6.88$	*161 'basset hound'* $d(J^{-1}(X_q), x_o) = \mathbf{7.52}$
			
754 'radio, wireless' $d(x_q, x_o) = 0.0$	*766 'rotisserie'* $d(x_q, x_o) = 0.17$	*754 'radio, wireless'* $d(J^{-1}(X_q), x_o) = 3.63$	*766 'rotisserie'* $d(J^{-1}(X_q), x_o) = 3.65$
			
626 'lighter, light' $d(x_q, x_o) = 0.0$	*470 'candle, taper'* $d(x_q, x_o) = 0.21$	*626 'lighter, light'* $d(J^{-1}(X_q), x_o) = 3.72$	*470 'candle, taper'* $d(J^{-1}(X_q), x_o) = 3.79$

Fig. 14. Examples of attacked images with spatial and JPEG75 quantizations. JPEG75 compression of the original image is also displayed in the third column.

recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
[25] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok, "Synthesizing robust adversarial examples," in *International conference on machine learning*. PMLR, 2018, pp. 284–293.
[26] A. Zandi, J.D. Allen, E.L. Schwartz, and M. Boliek, "Crew: Compression with reversible embedded wavelets," in *Proceedings DCC '95 Data Compression Conference*, 1995, pp. 212–221.
[27] Miska M. Hannuksela, Jani Lainema, and Vinod K. Malamal Vadakital, "The high efficiency image file format standard [standards in a nutshell]," *IEEE Signal Processing Magazine*, vol. 32, no. 4, pp. 150–156, 2015.

Benoit Bonnet graduated in aerospace engineering from the Ecole Polytechnique Féminine in Sceaux, France, in 2015. After working in the industry, he received the M.Sc. degree in image processing from Telecom ParisTech and Sorbonne University in 2019. He is currently pursuing a Ph.D. in security for artificial intelligence applications in Linkmedia team in Inria Rennes, France.

Teddy Furon (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in signal processing from Telecom ParisTech, in 1998 and in 2002, respectively. His research interests include the security related to multimedia, signal processing, and artificial intelligence. He has worked in industry (Thomson, Technicolor) and academia (Université catholique de Louvain, Belgium, and Inria Rennes, France, Linkmedia Team). He co-founded the company Imatag protecting rights of photo agencies. He has been an Associate Editor for four journals, including IEEE Transactions on Information Forensics and Security. He has been named the AID chair in Security of Artificial Intelligence.

Patrick Bas received the Electrical Engineering degree from the Institut National Polytechnique de Grenoble, France, in 1997, and then the Ph.D. degree in signal and image processing from Institut National Polytechnique de Grenoble, France, in 2000. He has co-organized the 2nd Edition of the BOWS-2 contest on watermarking in 2007, and the BOSS and Alaska contests on steganalysis respectively in 2010 and 2019.