Fooling an Automatic Image Quality Estimator

Benoit Bonnet[1], Teddy Furon[1], Patrick Bas[2]
[1]Univ. Rennes, Inria, CNRS, IRISA Rennes, France
[2]Univ. Lille, CNRS, Centrale Lille, UMR 9189, CRIStAL, Lille, France
contact:benoit.bonnet@inria.fr

ABSTRACT

This paper presents our work on the 2020 MediaEval task: "Pixel Privacy: Quality Camouflage for Social Images". Blind Image Quality Assessment (BIQA) is an algorithm predicting a quality score for any given image. Our task is to modify an image to decrease its BIQA score while maintaining a good perceived quality. Since BIQA is a deep neural network, we worked on an adversarial attack approach of the problem.

1 INTRODUCTION

The internet is flooded with images. This is especially true with the growth of social networks over the last decade. All this data is used to perform analysis to bring out new trends or to train predictive models. When it comes to images, deep neural networks vastly lead the landscape of machine learning. These deep neural networks are especially known to thrive on big datasets. This leads to the idea that more data leads to better models. While there certainly is truth to that affirmation, better learning mostly comes out of *better* data. Good data is data that both fits the task (*e.g.* people, places, objects detection) and whose quality is good. Due to the amount of available data, a human could not perform this cherry-picking of good data. Automated classifiers like BIQA [4] have been trained to assess the quality of an image. This classifier was trained on images whose quality was labeled based on the perceived quality of the media (*e.g.* resolution, compression artifacts). To protect one's data, images can be manipulated and slightly modified to defeat the automatic quality assessment [6]. We chose an adversarial attack approach to achieve this goal.

2 APPROACH

2.1 Adversarial Examples

Adversarial examples were first introduced by Szegedy *et al.* [8] in early 2014. They are usually studied in the case of image classification: An attack effectively crafts a perturbation of an image to a small extent but enough to fool even the best classifiers.

In this setup, an original image x_0 is given as an input to the trained neural network to estimate the probabilities $(\hat{p}_k(x_0))_k$ of being from class $k \in \{1, \dots, K\}$. The predicted class is given by:

$$\hat{c}(x_0) = \arg\max_k \hat{p}_k(x_0). \tag{1}$$

The classification is correct if $\hat{c}(x_0) = c(x_0)$ the ground truth class for x_0. The goal of an attack is to craft an imperceptible perturbation p such that the adversarial sample $x_a = x_0 + p$ verifies ideally:

$$x_a^\star = \arg\min_{x:\hat{c}(x)\neq c(x_0)} \|x - x_o\|, \tag{2}$$

Where $\|\cdot\|$ is a measure of distortion, in most cases the Euclidean distance. A small distortion makes it less likely for human to perceive that the image was manipulated.

BIQA is a deep neural network and as such is vulnerable to adversarial attacks. However BIQA is not a classifier returning a class prediction but a regressor giving a quality score $BIQA(x) \in [0, 100]$. The notion of adversarial sample thus needs to be redefined. In our case, we set a target score $s_a \in [0, 100]$. Regardless of the original score $BIQA(x_o)$, our adversarial sample now ideally verifies:

$$x_a^\star = \arg\min_{x:BIQA(x)<s_a} \|x - x_o\|, \tag{3}$$

2.2 Quantization

An original image x_0 in the spatial domain (*e.g.* PNG format) is a 3-dimensional discrete tensor: $x_0 \in \{0, 1, \dots, 255\}^n$ (with $n = 3 \times R \times C$, 3 color channels, R rows and C columns of pixels). The main objective of this task is to craft images: $x_a \in \{0, 1, \dots, 255\}^n$. This additional constraint to the attack is yet not easy to enforce.

In a deep neural network, this input image is first *preprocessed* onto a range domain that usually reduces variance of the data. Its purpose is to ease the learning phase and thus to increase the performance of a deep neural network. This *preprocessing* is defined by design before the training stage and cannot be modified at testing. In the case of BIQA, the range domain is $[-0.5, 0.5]^n$.

Most attacks of the literature are performed in this domain without consideration of the transformation it represents. This leads to an adversarial sample $x_a \in [0, 255]^n$ after reverting the preprocessing. To save this adversarial sample x_a as an image, the first step is then to round it which will erase most of the perturbation in the case of a low-distortion attack. Rounding is therefore likely to remove the adversarial property of the sample.

Paper [1] addresses this problem presenting a *post-processing* added on top of any attack to efficiently quantize a perturbation: It keeps the adversarial property while lowering the added distortion. The method is based on a classification loss to ensure adversariality defined as follows:

$$L(x) = \log(\hat{p}_{c(x_0)}(x)) - \log(\hat{p}_{\hat{c}(x)}(x)). \tag{4}$$

To adapt this method to the context of BIQA, we only need to redefine it to:

$$L(x) = BIQA(x) - s_a. \tag{5}$$

For a given x, $L(x) < 0$ ensures x scores under the target s_a.

MediaEval'20, December 14-15 2020, Online

3 EXPERIMENTAL WORK

In this task, we know the classifier (BIQA) and its parameters. We are therefore in a *white-box* setup. Most modern attacks are developed in this scenario, from the most basic FGSM [3] and IFGSM [5] to the most advanced PGD [7], C&W [2], BP [10]. FGSM is a non-iterative attack bringing a fast solution of the problem. Our work used this attack in the early stages as a proof of concept bringing a quick further understanding of the problem. Artifacts were visible. Instead all the results reported here are crafted using more the advanced PGD attack [7] in its L_2 optimization version. One input parameter is the distortion budget. We run the attack over 7 iterations with different distortion budgets (whose maximum value is set to 2000). A binary search quickly finds an adversarial sample with the lowest distortion.

3.1 JPEG compression

The final images will be evaluated on their JPEG [9] counterpart. This compression is done with a quality factor of 90. However there are many image compression sofwares providing different results. We used the command line `$ convert` to simulate this compression.

Tables 1 and 2 show for different methods both P_{PNG} and P_{JPEG} respectively the percentage of images successfully beating the target score in the PNG domain and the JPEG domain. Additionally Table 2 shows results of the jury as well.

3.2 Quantization

3.2.1 Spatial domain. The work [1] serves as a baseline for quantization. We only slightly adapt it as stated in Sect. 2.2. Table 1 reports our results for two target scores: $s_a = 30$ and $s_a = 50$. It appears that the perturbation crafted in the pixel domain is fragile when facing a JPEG compression.

3.2.2 DCT domain. The final image being evaluated after a JPEG compression, we explore a method adapting the quantization [1] to the DCT domain. Using the same notations [1]: Let X_o denote the image in the DCT domain, $X_a = X_o + P$ is the result of an initial attack like PGD, and $X_q = X_o + P + Q$ the final quantized transformed coefficients. We solve a Lagrangian formulation:

$$X_q = X_o + P + \arg\min_Q D(Q) + \lambda L(Q), \quad (6)$$

where λ is the Lagrangian multiplier controlling the tradeoff between the distortion $D(Q)$ and $L(Q)$ defined in (5). The distortion $D(Q)$ is defined as the squared L_2 norm of added perturbation: $D(Q) = \|\Delta \times (P+Q)\|^2$.

The quantization noise Q is s.t. $X_o + P + Q \in \Delta\mathbb{Z}^n$, where $\Delta \in \mathbb{N}^n$ is the quantization step matrix for JPEG QF=90. If we use a first order approximation of $L(Q)$, we can develop (6) in a second-degree polynomial function. For any coefficient j, this function is locally minimized by:

$$Q^\star(j) = -P(j) - \lambda \frac{G(j)}{2\Delta(j)}, \quad (7)$$

where $G = \nabla L(Q)|_{Q=\mathbf{0}}$ the gradient computed at $Q = 0$. This minimum however does not enforce $(P + Q^\star) \in \mathbb{Z}^n$. A simple rounding of $(P+Q)$ will then finalize the quantization. Finally we need to control a maximum allowed distortion. If λ gets big, $Q(j)$ become a very high value which is not desirable. The final value for the quantized perturbation in the DCT domain is thus bounded by $[-\frac{1}{\Delta}, \frac{1}{\Delta}]$. These images were submitted to the jury.

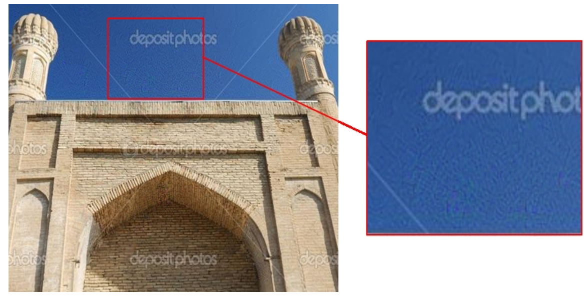

Figure 1: Image Places365_val_00019601c.png when quantized in the DCT domain at $s_a = 30$.

Table 1: Probabilities of success with a spatial Quantization

	P_{PNG}	P_{JPEG}
$s_a = 30$	**99.0**%	0.7%
$s_a = 50$	**100.0**%	11.1%

Table 2: Probabilities of success with a DCT Quantization

	P_{PNG}	P_{JPEG}	Accuracy after(JPEG90)	Number of times selected "best"
$s_a = 30$	77.5%	**63.8**%	23.82	40
$s_a = 50$	96.9 %	**91.6**%	0.91	57

4 RESULTS AND ANALYSIS

Tables 1 and 2 show the importance of considering the JPEG compression. When the image is quantized by the L_2 optimization in the spatial domain, most images will successfully be adversarial images. However, very few of them remain adversarial after the JPEG compression. The BIQA score on most images increases up to 10 points. If the quantization is done in the DCT domain, most of them remain adversarial and the task is successful. It is however obviously more difficult to beat a lower target score s_a. An interesting property of the DCT quantization is that it creates typical JPEG artifacts as seen on Figure 1. This is especially true in low frequency images since it is harder to remain undetectable in a such situation.

5 DISCUSSION AND OUTLOOK

The MediaEval task was a good opportunity to extend our previous work [1] to 1) a regressor BIQA, and 2) in the DCT domain. Saving the DCT coefficients directly into a JPEG image is more consistent as it offers a better control on adversariality. Another difficulty of this task was the lack of knowledge about the compression algorithm. We therefore worked in a 'gray' box setup. The results showed that JPEG compression have a big effect on the BIQA score of, at least, adversarial images (and probably any other quality estimator). Hopefully our JPEG compression is close to the one used in the contest which allowed transferability of our adversarial images.

REFERENCES

[1] Benoît Bonnet, Teddy Furon, and Patrick Bas. 2020. What If Adversarial Samples Were Digital Images?. In *Proc. of ACM IH&MMSec '20*. 55–66. https://doi.org/10.1145/3369412.3395062

[2] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *IEEE Symp. on Security and Privacy*.

[3] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *ICLR 2015, San Diego, CA, USA,*.

[4] V. Hosu, H. Lin, T. Sziranyi, and D. Saupe. 2020. KonIQ-10k: An Ecologically Valid Database for Deep Learning of Blind Image Quality Assessment. *IEEE Transactions on Image Processing* 29 (2020), 4041–4056.

[5] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017. Adversarial Machine Learning at Scale. (2017). arXiv:cs.CV/1611.01236

[6] Zhuoran Liu, Zhengyu Zhao, Martha Larson, and Laurent Amsaleg. 2020. Exploring Quality Camou□age for Social Images. In *Working Notes Proceedings of the MediaEval Workshop*.

[7] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR 2018, Vancouver, BC, Canada.*

[8] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. (2014). arXiv:cs.CV/1312.6199

[9] G. K. Wallace. 1992. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics* 38, 1 (1992), xviii–xxxiv. https://doi.org/10.1109/30.125072

[10] Hanwei Zhang, Yannis Avrithis, Teddy Furon, and Laurent Amsaleg. 2020. Walking on the Edge: Fast, Low-Distortion Adversarial Examples. *IEEE Transactions on Information Forensics and Security* (Sept. 2020). https://doi.org/10.1109/TIFS.2020.3021899