

# Photo Forensics from JPEG Dimples

Shruti Agarwal and Hany Farid  
Department of Computer Science, Dartmouth College  
{shruti.agarwal.gr, hany.farid}@dartmouth.edu

**Abstract**—Previous forensic techniques have exploited various characteristics of JPEG compression to reveal traces of manipulation in digital images. We describe a JPEG artifact that can arise depending on the choice of the mathematical operator used to convert DCT coefficients from floating-point to integer values. We show that the more commonly used floor or ceiling operators (but not the round operator) introduce a periodic artifact in the form of a single darker or brighter pixel—which we term a dimple—in  $8 \times 8$  pixel blocks. We describe the nature of this artifact, its prevalence in commercial cameras, and how this artifact can be quantified and used to detect a wide range of digital manipulations from content-aware fill to re-sampling, airbrushing, and compositing.

**Index Terms**—Image forensics, JPEG compression

## I. INTRODUCTION

Within the broad range of photo forensic techniques [1], format-based techniques exploit specific artifacts introduced by the underlying image format and compression scheme. Most notably, a variety of forensic techniques have been developed based on the JPEG compression format: the quantization values used to quantize the discrete cosine transform (DCT) coefficients are used to identify the recording camera manufacturer and model [2]–[4]; anomalies in the distribution of DCT coefficients are used to reveal multiple JPEG compressions [5]–[7]; and artifacts introduced by the JPEG blocking are used to detect localized tampering [8]–[10].

Adding to this body of format-based forensic techniques, we describe a lesser known JPEG artifact that arises from the choice of the mathematical operator—round, floor, or ceiling—used to convert DCT coefficients from floating-point to integer values. This artifact arises because the floor/ceiling operator, but not the round operator, uniformly decreases/increases the value of the quantized DCT coefficients. When the quantized DCT coefficients are converted back into the intensity-space, this uniform change manifests itself as a single darker or brighter pixel—which we term a dimple—in  $8 \times 8$  pixel blocks. Although this artifact has previously been noted [11], the root cause of the artifact has not been explained or exploited for forensic analyses.<sup>1</sup>

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA FA8750-16-C-0166). The views, opinions, and findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. We thank contributors to the Photographic Science and Technology forum of dpreview.com for helpful discussions. WIFS'2017, December, 4-7, 2017, Rennes, France. 978-1-5090-6769-5/17/\$31.00 ©2017 IEEE

<sup>1</sup>Jessica Fridrich and colleagues have previously observed this JPEG artifact and were actively eliminating it as it interfered with their PRNU analysis [12].

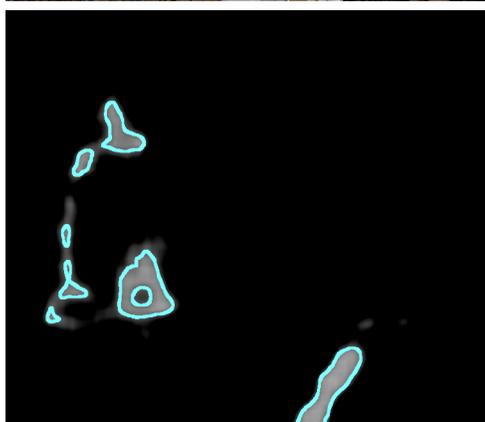


Fig. 1. A composite created by adding a sword and scale to Athena of Velletri (Louvre Museum, Paris). Shown in the lower panel is the result of our JPEG dimples analysis: the brighter regions correspond to detected manipulation. [Photo credits: Flickr user Marc Treble (scale) and www.armorymarek.com (sword).]

We describe the nature of this artifact, its prevalence in commercial cameras, its robustness to simple post-processing, and its efficacy in detecting a wide range of digital manipulations from content-aware fill to re-sampling, airbrushing, and compositing.

## II. JPEG DIMPLES

The JPEG encoding of a 3-channel RGB color image consists of seven basic steps: (1) convert from RGB to luminance/chrominance (YCbCr); (2) optionally subsample each channel by a factor of two or more (the chrominance channels are typically subsampled but the luminance channel is not); (3) partition each channel into non-overlapping  $8 \times 8$  pixel blocks; (4) convert the luminance values from unsigned to

signed integers (e.g., from  $[0, 255]$  to  $[-128, 128]$ ); (5) convert each block to the frequency domain using a 2-D discrete cosine transform (DCT); (6) quantize each DCT coefficient  $c$  by an amount  $q$  where the amount of quantization depends on the spatial frequency (the lower frequencies are typically quantized less than higher frequencies); (7) entropy encode the quantized DCT coefficients.

Of particular interest to us is the quantization in step 6. Of interest is not, as in [2], [3], the specific quantization values, but rather the operator for converting from a floating-point value to an integer. Specifically, there are three obvious ways in which a DCT coefficient  $c$  can be quantized by  $q$ : divide  $c$  by  $q$  and then apply the round, floor, or ceiling operator. Each of these operators have, of course, the same basic effect of converting a floating point number to an integer. Each operator, however, yields slightly different values. The floor operator, for example, will consistently yield slightly smaller values as compared to the original, while the ceiling operator will consistently yield slightly larger values. It is these differences, as we will describe next, that yield an artifact in images created by JPEG encoders that use the floor or ceiling operator, but not the round operator.

To see the nature of this artifact, consider the following 1-D example of quantization. Let  $\vec{s}$  be the following 1-D signal:

$$\vec{s} = (3.7 \ 8.3 \ 5.9 \ 5.1 \ 1.6 \ 2.4). \quad (1)$$

For simplicity, and because it is not critical to our analysis here, we will quantize this 1-D signal with  $q = 1$ . The quantized values, as computed with the round operator,  $\text{round}(\vec{s}/q) = [\vec{s}/q]$ , are:

$$\vec{s}_r = (4 \ 8 \ 6 \ 5 \ 2 \ 2) \quad (2)$$

And, the quantized values, as computed with the floor  $\lfloor \vec{s}/q \rfloor$  and ceiling operator  $\lceil \vec{s}/q \rceil$ , are:

$$\vec{s}_f = (3 \ 8 \ 5 \ 5 \ 1 \ 2) \quad (3)$$

$$\vec{s}_c = (4 \ 9 \ 6 \ 6 \ 2 \ 3). \quad (4)$$

In this toy — and admittedly contrived — example, the relationship between the three quantized signals and the original signal are:

$$\begin{aligned} \vec{s}_r &= \vec{s} + (0.3 \ -0.3 \ 0.1 \ -0.1 \ 0.4 \ -0.4) \\ \vec{s}_f &= \vec{s} + (-0.7 \ -0.3 \ -0.9 \ -0.1 \ -0.6 \ -0.4) \\ \vec{s}_c &= \vec{s} + (0.3 \ 0.7 \ 0.1 \ 0.9 \ 0.4 \ 0.6). \end{aligned}$$

Notice that the values in  $\vec{s}_r$  are intermittently larger or smaller than the original signal  $\vec{s}$ . On the other hand, the values in  $\vec{s}_f$  are consistently smaller than the original signal  $\vec{s}$  and the values in  $\vec{s}_c$  are consistently larger than the original signal. As a result, and to a first approximation, we can express the relationship between the results of the floor and ceiling operator as follows:

$$\vec{s}_f \approx \vec{s} - \alpha_f \vec{1} \quad (5)$$

$$\vec{s}_c \approx \vec{s} + \alpha_c \vec{1}, \quad (6)$$

where  $\vec{1} = (1 \ \dots \ 1)$  is a constant vector,  $\alpha_f$  is the mean of  $\vec{s}_f - \vec{s}$ , and  $\alpha_c$  is the mean of  $\vec{s}_c - \vec{s}$ .

Since this quantization is happening in the frequency domain, consider now the result of converting from the frequency domain back into the spatial domain:

$$D^{-1}(\vec{s}_f) = D^{-1}(\vec{s} - \alpha_f \vec{1}), \quad (7)$$

where  $D(\cdot)$  is the forward and  $D^{-1}(\cdot)$  is the inverse DCT. Because of the linearity of the DCT, the right-hand side of this equation can be expressed as:

$$\begin{aligned} D^{-1}(\vec{s}_f) &= D^{-1}(\vec{s}) - \alpha_f D^{-1}(\vec{1}) \\ D^{-1}(\vec{s}_f) &= D^{-1}(\vec{s}) - \alpha_f \vec{\delta}, \end{aligned} \quad (8)$$

where the inverse DCT of a constant signal,  $\vec{1}$ , is an impulse  $\vec{\delta}$ . The ceiling operator yields a similar result except that the impulse is now additive.

$$\begin{aligned} D^{-1}(\vec{s}_c) &= D^{-1}(\vec{s} + \alpha_c \vec{1}) \\ D^{-1}(\vec{s}_c) &= D^{-1}(\vec{s}) + \alpha_c D^{-1}(\vec{1}) \\ D^{-1}(\vec{s}_c) &= D^{-1}(\vec{s}) + \alpha_c \vec{\delta}. \end{aligned} \quad (9)$$

In this toy example, the result of quantizing with the floor function is that the left-most<sup>2</sup> value in  $\vec{s}_f$  will be slightly smaller than in  $\vec{s}$  due to the subtraction of an impulse while the left-most value in  $\vec{s}_c$  will be slightly larger due to the addition of the impulse.

In the full 2-D case, this process is repeated for every  $8 \times 8$  pixel block leading to a periodic artifact in which the top-left corner of each block is consistently dark (floor) or light (ceiling). We informally refer to this artifact as JPEG dimples.

The synthetically-generated fractal<sup>3</sup> image in Fig. 2 was compressed using a custom JPEG coder with either the round, floor, or ceiling operator. Shown in each column is the intensity image (top) and a magnified view of the upper-left corner of the image (bottom). The JPEG dimples are clearly seen in the magnified images. Notice also that, as predicted, the dimples occur in every  $8 \times 8$  block and are darker for the floor operator and brighter for the ceiling operator, but are not introduced by the round operator.

We will show that many cameras introduce this artifact. First, however, we describe a simple technique for automatically detecting the presence of JPEG dimples.

#### A. Automatic Detection

We use a template-based approach to detect JPEG dimples. A dimple can be approximated with an impulse in the upper left corner of each  $8 \times 8$  pixel block. We, therefore, define a template  $T(x, y)$  of the same size as the image  $I(x, y)$  in question. This template has a value of 0 everywhere except

<sup>2</sup>The inverse DCT of a constant signal  $\vec{1}$  is an impulse  $\vec{\delta}$ . The location of this impulse in the spatial domain is dictated by the phase of the constant signal in the frequency domain. In our case, this phase is zero and so the impulse is positioned at the left-most sample.

<sup>3</sup>A fractal image is created in the Fourier domain with a  $1/w$  magnitude and random phase.

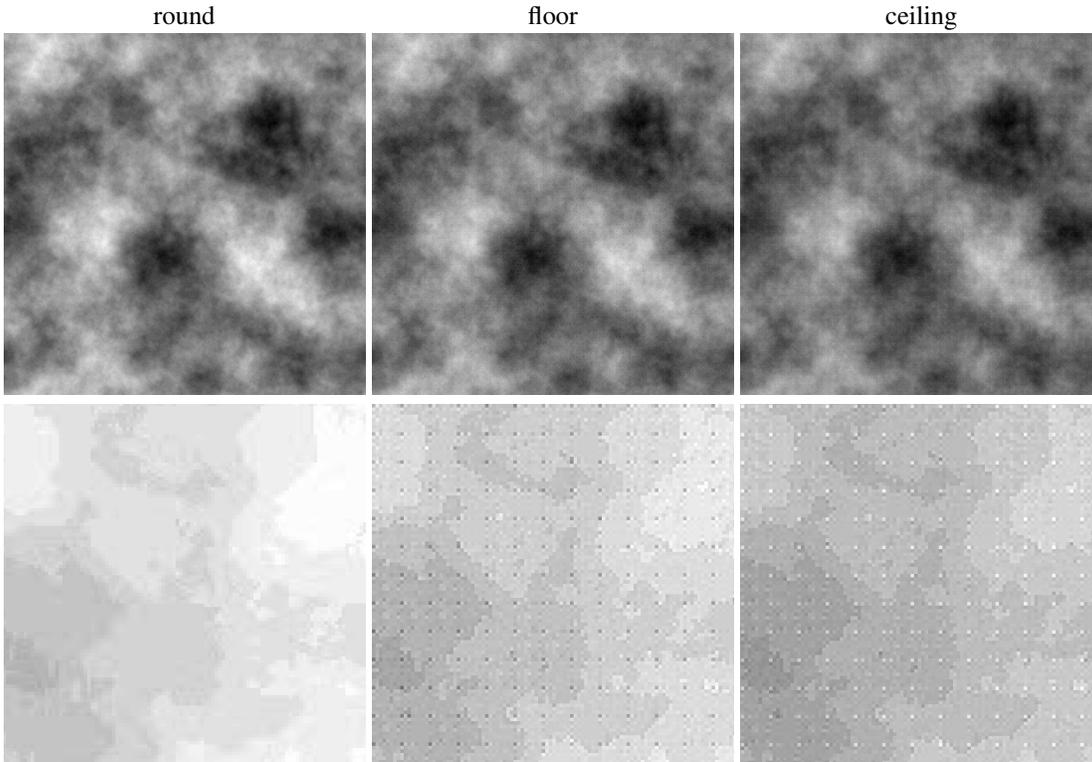


Fig. 2. Shown from left to right are synthetically-generated images that have been JPEG compressed using one of three operators to quantize the DCT coefficients: round, floor, or ceiling. Shown in the bottom row is a magnified view of the upper left corner of each image. The periodic JPEG dimples—a single dark or light pixel in the upper left corner of each  $8 \times 8$  pixel block—are introduced by the floor and ceiling operator, but not the round operator.

for a value of 1 in the upper left corner of every  $8 \times 8$  pixel block.

The strength of the JPEG dimples is measured using the peak to correlation energy (PCE) [12]. The larger the PCE value the more prominent the dimples. The PCE,  $p_I$ , for an image  $I(x, y)$  is computed as follows:

$$p_I = \frac{F_I^2(\hat{u}, \hat{v})}{\frac{1}{63} \sum_{(u,v) \neq (\hat{u}, \hat{v})} F_I^2(u, v)}, \quad (10)$$

where,

$$F_I(u, v) = \sum_x \sum_y I(x, y) T(x + u, y + v), \quad (11)$$

where it is assumed that  $I(\cdot)$  and  $T(\cdot)$  are each zero-mean and unit-sum, and where  $u, v \in [0, 7]$  and  $(\hat{u}, \hat{v})$  corresponds to the offset that maximizes  $F_I(\cdot)$ :

$$(\hat{u}, \hat{v}) = \arg \max_{u, v} (F_I^2(u, v)). \quad (12)$$

Due to the periodicity of the template, the spatial offsets,  $(u, v)$  range over a single  $8 \times 8$  pixel block.

We note that the PCE will respond regardless of where in the  $8 \times 8$  block the impulse appears. This allows us to handle both portrait and landscape images (i.e.,  $90^\circ$  rotations) as well as image cropping.

We have found that the underlying image content can sometimes interfere with the detectability of JPEG dimples.

To contend with this, we apply a  $3 \times 3$  Wiener filter to each RGB channel and average the resulting noise residual across all three channels. To further suppress image content, non-overlapping blocks of size  $32 \times 32$  are then averaged across the entire image (or, optionally, a portion of the image). The PCE is then computed against this averaged  $32 \times 32$  block and the corresponding template of the same size.

### B. Prevalence

In order to determine the prevalence of JPEG dimples in commercial cameras, we collected from Flickr approximately 40,000 unmodified images [4]. These images spanned a total of 4,039 camera configurations — defined as unique camera manufacturer, model, and capture resolution.

In the first of two analyses we considered 1,017 of 4,039 configurations each with 5 to 20 images recorded at the maximum resolution afforded by the camera (as determined by dpreview.com). The presence of dimples for a camera configuration is determined by averaging the PCE (Section II-A) across all available images for the camera configuration. A camera configuration is said to introduce dimples if the average PCE is greater than an empirically determined value of 13. Shown in Fig. 3 is the prevalence of dimples for each of 31 different camera manufacturers. Overall, 67% of 1,017 camera configurations contain the JPEG dimple artifact. Images from Asus, HTC and Sony consistently contain dimples regardless of the camera model. A few other manufacturers (e.g., Apple,

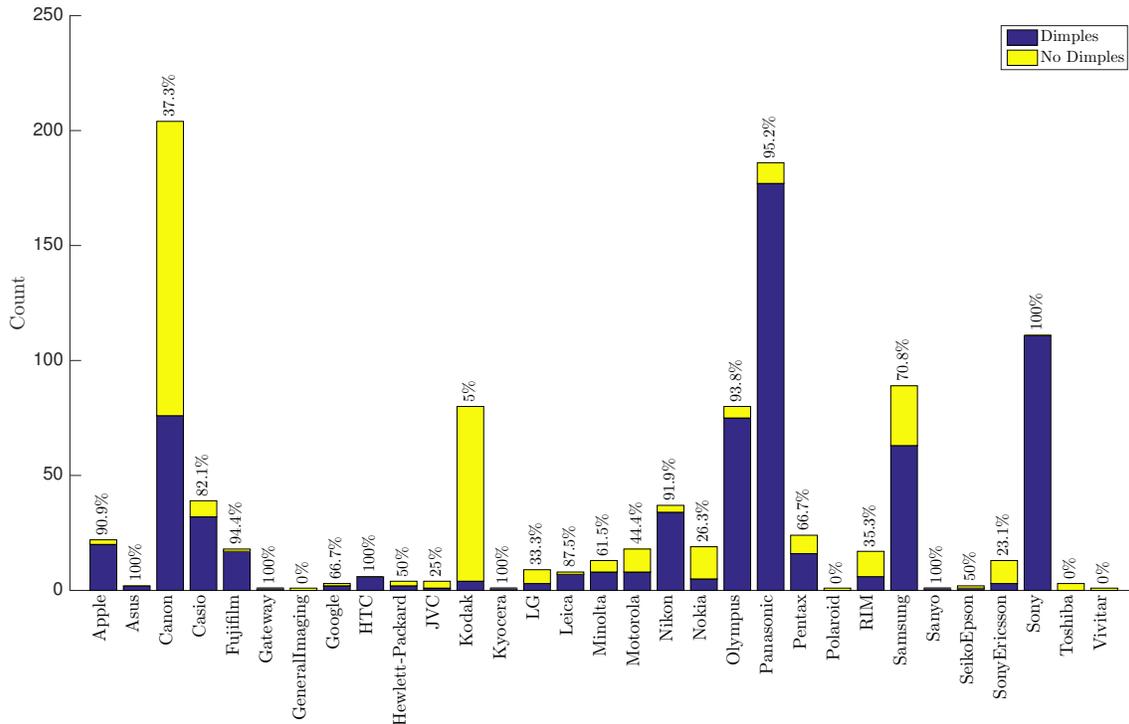


Fig. 3. The prevalence of JPEG dimples per camera manufacturer. Each bar corresponds to the total number of models per camera manufacturer. The portion of each bar shaded blue/yellow corresponds to the models with/without dimples. The numeric value above each bar corresponds to the percentage of models with dimples.

Fujifilm, Nikon, Olympus, and Panasonic) are also almost perfectly consistent except for a small number of models. On the other hand, images from Kodak cameras almost never contain dimples. In between these extremes are, for example, Canon and Samsung in which the presence depends on the specific camera model.

In the second analysis, we observe that the strength of the dimples varies across camera manufacturers. Shown in Fig. 4 is the average PCE for 19 of the 31 manufacturers (Fig. 3) which introduce dimples. The average PCE ranges from a maximum of 39 (Sony) to a minimum of 16 (Canon). The cause of these differences across manufacturers is not immediately clear, however, the differences appear to be unrelated to image resolution or quality.

### III. FORENSIC APPLICATIONS

In this section, we show that some common forms of manipulation can destroy JPEG dimples. We also show how to locally detect the presence of JPEG dimples in order to detect these manipulations. Shown in Fig. 6(a) and (c) are original and manipulated images. From top to bottom, the manipulations are: insertion of the snow leopard, removal of a person using content-aware fill, rotation of the stop sign, subtle (almost imperceptible) airbrushing of the face to remove small blemishes, geometric warping of the lion’s face, and modification of the police-car hood.

In order to detect the type of local manipulations shown in Fig. 6, the measure of JPEG dimples estimated from an entire

image is now applied to overlapping  $512 \times 512$  windows to yield a prominence map that specifies the per-pixel dimple strength. Each pixel of a prominence map is computed by averaging the PCE over all  $512 \times 512$  windows that contains the pixel.

Shown in Fig. 6(b) and (d) are the per-pixel prominence maps for the original and manipulated images. The intensity in these maps correspond to PCE value in the range of 0 (white) to 20 (black) where all values greater than 20 are, for visual clarity, clamped to black. The cyan contour lines correspond to a PCE value of 13 and values within this closed contour correspond to PCE values less than 13. All of these parameters were held constant across the first five examples in Fig. 6. In the sixth and last example of this figure, the prominence map was computed over a  $256 \times 256$  window. The smaller window was necessary in order to detect the relatively small bounding box of the manipulation. In all cases, the manipulation destroys the JPEG dimples which is then easily detected (see also Fig. 1).

#### A. Resilience to post-processing

In this section, we explore the resilience of JPEG dimples to common forms of post-processing: double JPEG compression, gamma correction, additive noise, and scaling. We analyzed 2,000 images from various manufacturers — Asus, Canon, Casio, HTC, HP, LG, Leica, Minolta, Motorola, Olympus, Panasonic, Samsung, and Sony. The average PCE value of these 2,000 images, before any post-processing, is 46.

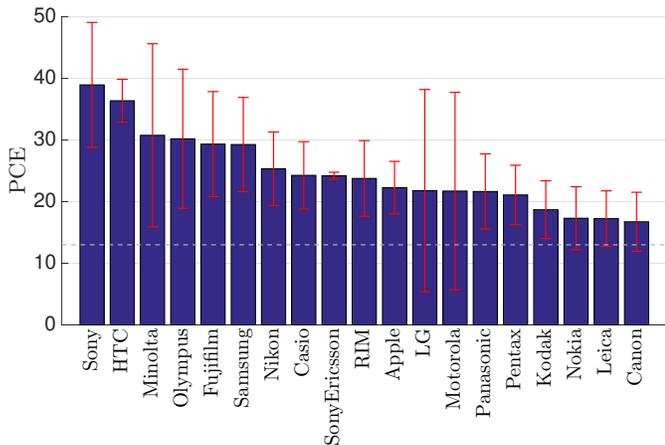


Fig. 4. The average strength of JPEG dimples per camera manufacturer. Each bar corresponds to the average PCE value for all available models per manufacturer. The dashed horizontal line corresponds to our detection threshold. The error bars correspond to plus/minus one standard deviation.

Each image is post-processed using one of the four manipulations. For double JPEG compression, the second compression quality ranged from 10 to 100 (per MatLab’s JPEG compression setting). For gamma correction, the exponent ranged from 0.2 to 1.8. For additive noise, the SNR of additive Gaussian noise ranged from  $-20$  to  $20$ db, and for scaling, the scale factor ranged from 0.5 to 1.0 (scaling was performed using bicubic interpolation).

Shown in Fig. 5 is the resilience of the JPEG dimples to each of these manipulations. The detection is resilient to most of these manipulations except for JPEG quality of 30 and less (out of 100), additive noise of  $-10$ db and less, and scaling of less than 0.6.

#### IV. DISCUSSION

We have explored, explained, and exploited a JPEG artifact—JPEG dimples—for the purpose of photo forensics. We argue that the choice of mathematical operator—round, floor, or ceiling—used to quantize the DCT coefficients is the cause of this artifact. We have provided a technique that can be used to locally and globally detect JPEG dimples. Our analysis of 1,017 commercial cameras from 31 different manufacturers reveals the wide-spread prevalence of JPEG dimples. We have shown that JPEG dimples can be used to reliably detect a wide range of manipulations from content-aware fill to re-sampling, airbrushing and compositing. And, we have shown that the JPEG dimples are resilient to simple post-processing.

We have provided an algebraic explanation that correctly predicts the presence and absence of JPEG dimples. Further, we have provided empirical evidence of the presence of this artifact in images from commercial devices. There are, however, some unanswered questions that will require further study. First, it is not clear to us why the strength of the JPEG dimples vary, by more than a factor of two, across manufacturers. Second, in all of the images collected from commercial devices, we find that the JPEG dimples manifest

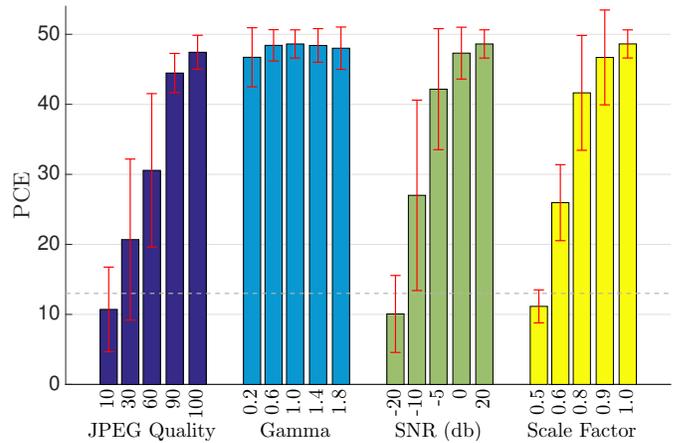


Fig. 5. The average strength of JPEG dimples as a function of double JPEG compression quality, gamma correction, signal-to-noise (SNR), and scaling. The dashed horizontal line corresponds to our detection threshold. The error bars correspond to plus/minus one standard deviation.

as a brighter pixel corresponding to the use of the ceiling operator (as opposed to a darker pixel corresponding to the floor operator). It is not clear why the ceiling operator is more uniformly used.

Lastly, we note that Adobe Photoshop does not introduce JPEG dimples (nor does MatLab). This means that if an image with JPEG dimples is edited and saved from Photoshop, the dimples will not be re-introduced.

#### REFERENCES

- [1] H. Farid, *Photo Forensics*. MIT Press, 2016.
- [2] —, “Digital image ballistics from JPEG quantization,” Department of Computer Science, Dartmouth College, Tech. Rep. TR2006-583, 2006.
- [3] J. D. Kornblum, “Using JPEG quantization tables to identify imagery processed by software,” *Digital Investigation*, vol. 5, pp. S21–S25, 2008.
- [4] E. Kee, M. K. Johnson, and H. Farid, “Digital image authentication from JPEG headers,” *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1066–1075, 2011.
- [5] A. C. Popescu and H. Farid, “Statistical tools for digital forensics,” in *Proceedings of the 6th International Conference on Information Hiding*, 2004, pp. 128–147.
- [6] J. He, Z. Lin, L. Wang, and X. Tang, “Detecting doctored JPEG images via DCT coefficient analysis,” in *European Conference on Computer Vision*, Graz, Austria, 2006.
- [7] B. Mahdian and S. Saic, “Detecting double compressed JPEG images,” in *3rd International Conference on Crime Detection and Prevention*, 2009, pp. 1–6.
- [8] S. Ye, Q. Sun, and E. Chang, “Detecting digital image forgeries by measuring inconsistencies of blocking artifact,” in *2007 IEEE International Conference on Multimedia and Expo*, 2007, pp. 12–15.
- [9] W. Luo, Z. Qu, J. Huang, and G. Qiu, “A novel method for detecting cropped and recompressed image block,” in *IEEE Conference on Acoustics, Speech and Signal Processing*, 2007, pp. 217–220.
- [10] T. Bianchi and A. Piva, “Image forgery localization via block-grained analysis of jpeg artifacts,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1003–1017, 2012.
- [11] Y. L. Lee, H. C. Kim, and H. W. Park, “Blocking effect reduction of JPEG images by signal adaptive filtering,” *IEEE Trans. Image Processing*, vol. 7, no. 2, pp. 229–234, 1998.
- [12] M. Goljan, J. Fridrich, and T. Filler, “Large scale test of sensor fingerprint camera identification,” *Proc. SPIE*, vol. 7254, pp. 72 540I–72 540I–12, 2009.
- [13] M. A. Robertson and R. L. Stevenson, “DCT quantization noise in compressed images,” *IEEE Trans. Circuits Syst. Video Techn.*, vol. 15, no. 1, pp. 27–38, 2005.

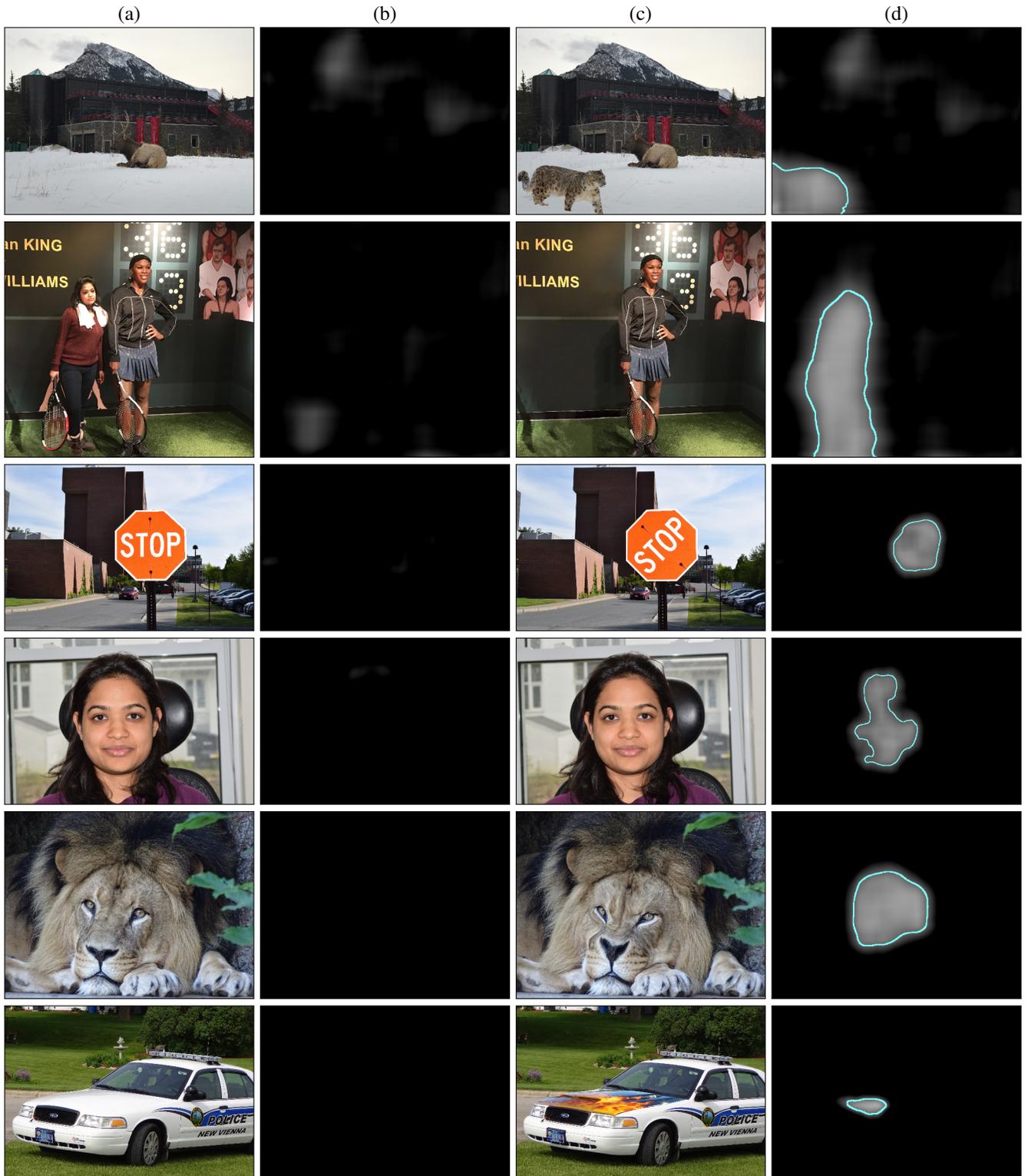


Fig. 6. Detecting manipulation from JPEG dimples. Shown in columns (a) and (c) are the original and manipulated images. The images, from top to bottom, were photograph with an iPhone 7, iPhone 6s, Nikon D3300, Nikon D3300, Sony DSC-HX400V, and Sony DSC-WX220. Shown in columns (b) and (d) are the prominence maps giving the strength of the local JPEG dimples. The intensity in these maps correspond to PCE value where black corresponds to a high PCE value (dimples are present) and white corresponds to a low PCE value (dimples are not present). The cyan contour lines correspond to a PCE value of 13 and values within this closed contour correspond to PCE values less than 13. [Photo credits: snow leopard (first row): [www.andbeyond.com](http://www.andbeyond.com); lion and police car (last two rows): [www.flickr.com](http://www.flickr.com); and car hood (last row): [www.allfordmustangs.com](http://www.allfordmustangs.com)]