Focus Manipulation Detection via Photometric Histogram Analysis

Can Chen
Computer Information & Science
University of Delaware
Newark, DE 19716
canchen@udel.edu

Scott McCloskey
Honeywell ACS Labs
Golden Valley, MN 55422
scott.mccloskey@honeywell.com

Jingyi Yu
University of Delaware, ShanghaiTech University
Shanghai, CHN 200031
yu@eecis.udel.edu

Abstract

With the rise of misinformation spread via social media channels, enabled by the increasing automation and realism of image manipulation tools, image forensics is an increasingly relevant problem. Classic image forensic methods leverage low-level cues such as metadata, sensor noise fingerprints, and others that are easily fooled when the image is re-encoded upon upload to facebook, etc. This necessitates the use of higher-level physical and semantic cues that, once hard to estimate reliably in the wild, have become more effective due to the increasing power of computer vision. In particular, we detect manipulations introduced by artificial blurring of the image, which creates inconsistent photometric relationships between image intensity and various cues. We achieve 98% accuracy on the most challenging cases in a new dataset of blur manipulations, where the blur is geometrically correct and consistent with the scene's physical arrangement. Such manipulations are now easily generated, for instance, by smartphone cameras having hardware to measure depth, e.g. 'Portrait Mode' of the iPhone7Plus. We also demonstrate good performance on a challenge dataset evaluating a wider range of manipulations in imagery representing 'in the wild' conditions.

1. Introduction

Image forensics is a long-standing problem whose objective is to detect manipulations which call into question the veracity of the image content. Manipulating images is now easier than ever, given the computer vision powered advance of software packages such as PhotoShop, and the forensic task is harder than ever due to sheer volume of images created and shared on modern platforms. A false detection rate that would have been acceptable in a previous era is now completely unworkable, for example, when applied to the 30 billion images shared via Instagram.

Artificial blur can be used to obscure important details and, unlike metadata edits or other forensic cues, both motion and optical blur are naturally-occurring phenomena. Indeed, as shown by the recent popularity of Portrait Mode-type features, blur is a *desirable* feature of high quality images. Shallow depth of field (DoF), in which non-subject parts of the image are blurred, is a signature element in professional photography and film editing. DoF effects are known to improve photorealism, mediate monocular depth perception [21, 22], and improve the salience of focused objects. Shallow DoF with smooth Bokeh and correct blur in out-of-focus regions has long required high-end Digital Single-Lens Reflex (DSLR) cameras with high quality lenses, while small aperture mobile phone cameras sharply image all parts of the scene. Not surprisingly, then, enabling shallow DoF effects have been identified as a key objective of computational photography teams at Google [2] and elsewhere. Software and mobile phone apps such as PhotoShop [1], FabFocus [4], Depth Effects [3], etc. provide the shallow DoF effect, but require a lot of user effort and don't enforce 3D geometric consistency. To address this, recent smartphones - such as the iPhone7Plus, Google Pixel 2, and HuaweiHonor8 - use 3D sensing hardware and/or algorithms which automates the production of shallow DoF images that are geometrically correct.

Though these manipulations are geometrically consistent and visually persuasive, there are still detectable differences with images having genuinely shallow DoF. The key commonality of these manipulation methods is that they apply the local blur in software, where the blur kernel doubles as a de-noising filter. As a result, these manipulations in-

troduce photometric inconsistencies in noise levels, which are the primary cue for our detection. In addition, we use JPEG double quantization and demosaicing cues to detect more general focus manipulations. In this paper, we present a photo forensic method to distinguish images having a naturally shallow DoF from manipulated ones, by integrating a number of cues under a fusion of two deep convolution networks with small receptive fields for histogram classification. Comprehensive experiments show that our learning-based approach outperforms existing solutions on *both* a newly-collected dataset of Portrait Mode-type imagery and a public challenge dataset including a wider range of manipulations in imagery representing 'in the wild' conditions.

2. Related Work

Our primary contribution is to the field of image forensics, which is a longstanding research area reviewed comprehensively by Qureshi and Deriche [24]. Within forensics, blur has been used as a cue to detect splicing operations, where the spliced-in object(s) may be inconsistently blurred with respect to other parts of the image, or may have boundaries which are implausible. Chen *et al*. [12] recently introduced a method that detects such implausible boundaries by classifying edge patches from an image, but that method's dependence on pixel-wise statistics near edges leads to problems on compressed 'in the wild' imagery. Earlier, Bahrami *et al*. [6, 7] partially automated the forensic task by segmenting and labeling of image regions based on local blur; the ultimate classification of an image as manipulated or authentic is left to a human observer, though, so the method doesn't address web-scale forensics. Even if the final classification were automated, though, segment-level labeling of the type and extent of blur wouldn't address the geometrically correct manipulations that we do. A recent post by Google [2] explains the Pixel 2 portrait mode implementation in some detail, but other companies are less forthcoming on their methods. Differences between these implementations, like whether the de-blurring is applied before or after compression, have a significant impact on our ability to detect focus manipulations.

Another important consideration is how the spatially-variant blur is applied, for which there is extensive past research. Classic methods simulate blur by spatially convolving (filtering) neighbor pixels within a synthetic blur kernel. Starting from iterative filtering [26], algorithms in this vein have evolved to more sophisticated solutions including pre-blurring [25, 14, 15], anisotropic diffusion [10, 23], separable Gaussian filters [25, 30], etc. To mitigate visual artifacts around edges, multiple layer approaches have become increasingly popular. Inspired by an earlier object-space grouping method [27], more recent approaches [16, 17, 9, 8, 18] decompose a pinhole image into several sub-images according to the depth of pixels; each sub-image is then separately filtered via either single layer scattering [16, 17], Fast Fourier Transforms [9, 8], or customized pyramidal processing [18].

3. Primary Cue: Image Noise

Regardless of how the local blurring is implemented, the key difference between optical blur and portrait mode-type processing can be found in image noise. Fig. 1 shows the digital camera imaging process, along with a description of various noise sources. When blur happens optically, before photons reach the sensor, only small signal-dependent noise impacts are observed. When blur is applied algorithmically to an already digitized image, however, the smoothing or filtering operation also implicitly de-noises the image. Since the amount of denoising is proportional to the amount of local smoothing or blurring, and since we are interested in spatially non-uniform blurring operations, differences in the amount of algorithmic local blur can be detected via inconsistencies between the local intensity and noise level. Two regions of the image having approximately the same intensity should also have approximately the same level of noise. If one region is blurred more than the other, or one is blurred while the other is not, an inconsistency is introduced between the intensities and local noise levels.

For our noise analysis, we extend the combined noise models of [29, 19]. Ideally, a pixel produces a number of electrons E_{num} proportional to the average irradiance from the object being imaged. However, shot noise N_S is a result of the quantum nature of light and captures the uncertainty in the number of electrons stored at a collection site; N_S can be modeled as Poisson noise. Additionally, site-to-site non-uniformities called *fixed pattern noise* K are a multiplicative factor impacting the number of electrons; K can be characterized as having mean 1 and a small spatial variance σ_K^2 over all of the collection sites. Thermal energy in silicon generates free electrons which contributes *dark current* to the image; this is modeled as an additive factor N_{DC}, modeled as Gaussian noise. The on-chip output amplifier sequentially transforms the charge collected at each site into a measurable voltage with a scale A, and the amplifier generates zero mean read-out noise N_R with variance σ_R^2. De-mosaicing is applied in color cameras to interpolate two of the 3 colors at each pixel, and introduces an error which is sometimes modeled as noise. After this, the camera response function (CRF) $f(\cdot)$ maps this voltage via a non-linear transform to improve perceptual image quality. Lastly, the analog-to-digital converter (ADC) approximates the analog voltage as an integer multiple of a quantization step q. The quantization noise can be modeled as the addition of a noise source N_Q.

With these noise sources in mind, we can describe a dig-

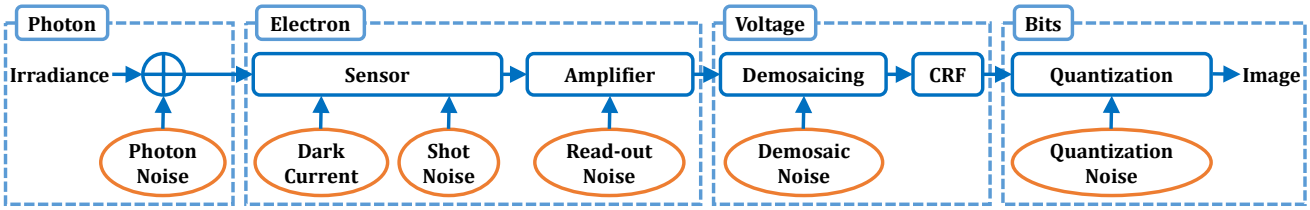


Figure 1. Digital camera imaging pipeline, showing the various sources of noise in a captured image.

itized 2D image as:

$$D(x,y) = f\Big((K(x,y)E_{num}(x,y) + N_{DC}(x,y) + N_S(x,y) + N_R(x,y))A\Big) + N_Q(x,y) \quad (1)$$

The variance of the noise is given by

$$\sigma_N^2(x,y) = f'^2\Big(A^2\big(K(x,y)E_{num}(x,y) + \mathrm{E}[N_{DC}(x,y)] + \sigma_R^2\big)\Big) + \frac{q^2}{12} \quad (2)$$

where $\mathrm{E}[\cdot]$ is the expectation function. This equation tells us two things which are typically overlooked in the more simplistic model of noise as an additive Gaussian source:

1. The noise variance's relationship with intensity reveals the shape of the CRF's derivative f'.

2. Noise has a signal-dependent aspect to it, as evidenced by the E_{num} term in (2).

An important corollary to this is that different levels of noise in regions of an image having different intensities is not *per se* an indicator of manipulation, though it has been taken as one in past work [20]. We show in our experiments that, while the noise inconsistency cue from [20] has some predictive power in detecting manipulations, a proper accounting for signal-dependent noise via its relationship with image intensity significantly improves detection performance.

Measuring noise in an image is, of course, ill-posed, and is equivalent to the long-standing image de-noising problem. For this reason, we leverage three different approximations of local noise, measured over approximately-uniform image regions: intensity variance, intensity gradient magnitude, and the noise feature of [20] (abbreviated NOI). Each of these is related to the image intensity of the corresponding region via a 2D histogram. This step translates subtle statistical relationships in the image to shape features in the 2D histograms which can be classified by a neural network. As we show in the experiments, our detection performance on histogram features significantly improves on that of popular approaches applied directly to the pixels of the image.

Figure 2. 4 Different Bayer CFA patterns.

4. Additional Photo Forensic Cues

One of the key challenges in forensic analysis of images 'in the wild' is that compression and other post-processing may overwhelm subtle forgery cues. Indeed, noise features are inherently sensitive to compression which, like blur, smooths the image. In order to improve detection performance in such challenging cases, we incorporate additional forensic cues which improve our method's robustness. Some manipulation methods operate on a JPEG image, i.e. after all steps of the pipeline; others operate on raw images, and apply steps like the CRF after the manipulation. Some require user intervention to determine the amount of local blurring; others, including portrait modes, use hardware to measure depth, and obviate the need for user intervention. As such, there are a range of different cues that can reveal manipulations in a subset of the data.

4.1. Demosaicing Artifacts

At a sensor pixel which only records one of the red, green, or blue channels of the image, the remaining two colors must be interpolated from neighboring pixels which measure them. Forensic researchers have shown that the differences between the de-mosaicing algorithm, and the differences between the physical color filter array bonded to the sensor, can be detected from the image. Since focus manipulations are applied on the demosaiced images, the local smoothing operations will alter these subtle Color Filter Array (CFA) demosaicing artifacts. In particular, the lack of CFA artifacts or the detection of weak, spatially-varying CFA artifacts indicates the presence of global or local tampering, respectively.

Following the method of [13], we consider the demosaicing scheme f_d being bilinear interpolation. We divide the image into $W \times W$ sub-blocks, and only compute the demosaicing feature at the non-smooth blocks of pixels. Denote each non-smooth block as B_i, where $i = 1, \ldots, m_B$, and m_B is the number of non-smooth blocks in the image. We use the four different Bayer pattern CFA arrangements shown in Fig. 2, and assess the error between the

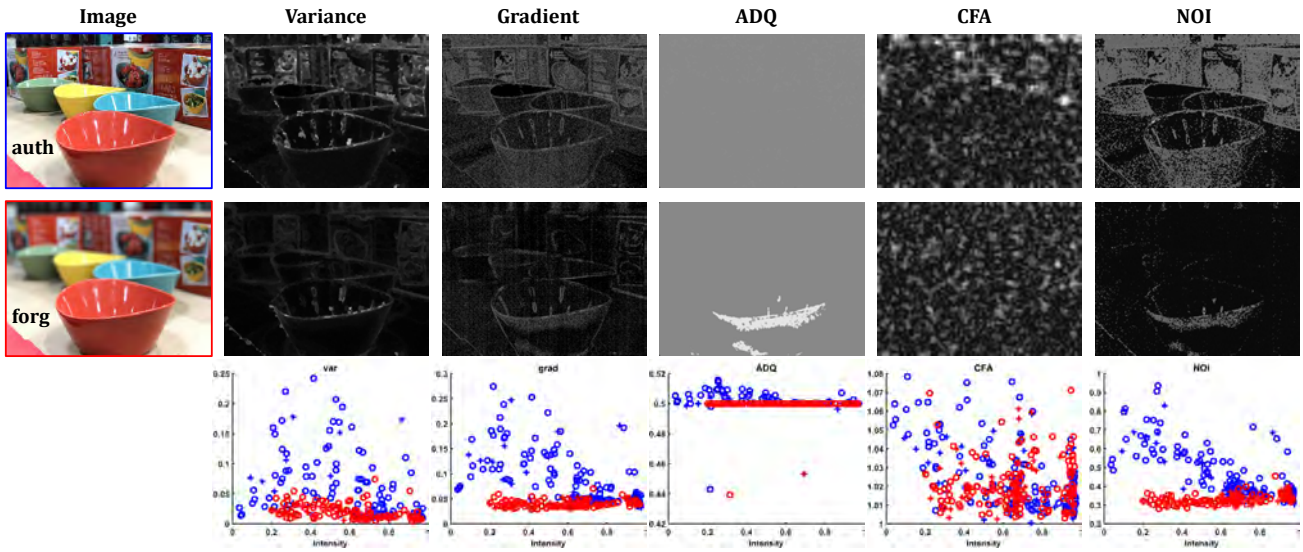


Figure 3. Feature maps and histogram for authentic and manipulated images. On the first row are the authentic image with feature maps; the second row shows the corresponding maps for the manipulated image. We show scatter plots relating the features to intensity in the third row, where blue sample points correspond to the authentic image, and red corresponds to a manipulated DoF image (which was taken with an iPhone).

image's pixel values and re-interpolated intensities assuming each of the four patterns. The re-interpolation error of i-th sub block for the k-th CFA pattern θ_k is defined as: $\hat{B}_{i,k} = f_d(B_i, \theta_k)$ and $k = 1, \ldots, 4$. The MSE error matrix $E_i^{(2)}(k,c), c \in R, G, B$ between the blocks B and $\hat{B}$ is computed in non-smooth regions all over the image. Therefore we define the metric to estimate the uniformity of normalized green channel column vector as

$$F = median\left(\sum_{l=1}^{4} |100 \times \frac{E_i^{(2)}(k,2)}{\sum_{l=1}^{3} E_i^{(2)}(l,2)} - 25|\right)$$
$$E_i^{(2)}(k,c) = 100 \times \frac{E_i(k,2)}{\sum_{l=1}^{3} E_i(l,2)}$$
$$E_i(k,c) = \frac{1}{W \times W} \sum_{x=1}^{W} \sum_{y=1}^{W} \left(B_i(x,y,c) - \hat{B}_{i,k}(x,y,c)\right)^2 \quad (3)$$

4.2. JPEG Artifact

In some portrait mode implementations, such as the iPhone, the option to save both an original and a portrait mode image of the same scene suggests that post-processing is applied *after* JPEG compression. Importantly, both the original JPEG image and the processed version are saved in the JPEG format *without resizing*. Hence, Discrete Cosine Transform (DCT) coefficients representing unmodified areas will undergo two consecutive JPEG compressions and exhibit double quantization (DQ) artifacts, used extensively in the forensics literature. DCT coefficients of locally-blurred areas, on the other hand, will result from non-consecutive compressions and will present weaker artifacts.

We follow the work of [11], and use Bayesian inference to assign to each DCT coefficient a probability of being doubly quantized. Accumulated over each 8×8 block of pixels, the DQ probability map allows us to distinguish original areas (having high DQ probability) from tampered areas (having low DQ probability). The probability of a block being tampered can be estimated as

$$p = 1/\left(\prod_{i|m_i \neq 0} \left(R(m_i) - L(m_i)\right) * k_g(m_i) + 1\right)$$
$$R(m) = Q_1\left(\left\lceil \frac{Q_2}{Q_1}\left(m - \frac{b}{Q_2} - \frac{1}{2}\right)\right\rceil - \frac{1}{2}\right)$$
$$L(m) = Q_1\left(\left\lfloor \frac{Q_2}{Q_1}\left(m - \frac{b}{Q_2} + \frac{1}{2}\right)\right\rfloor + \frac{1}{2}\right) \quad (4)$$

where m is the value of the DCT coefficient. $k_g(\cdot)$ is a Gaussian kernel with standard deviation σ_e/Q_2. Q_1, Q_2 are the quantization steps used in the first and second compression, respectively. b is the bias.

5. Focus Manipulation Detection

To summarize the analysis above, we adopt 5 types of features: color variance (VAR), image gradient (GRAD), double quantization (ADQ) [11], color filter artifacts (CFA) [13] and noise inconsistencies (NOI) [20] for refocusing detection. Each of these features is computed densely at each location in the image, and Fig. 3 illustrates the magnitude of these features in a feature map for an authentic image (top row) and a portrait mode image (middle row). Though there

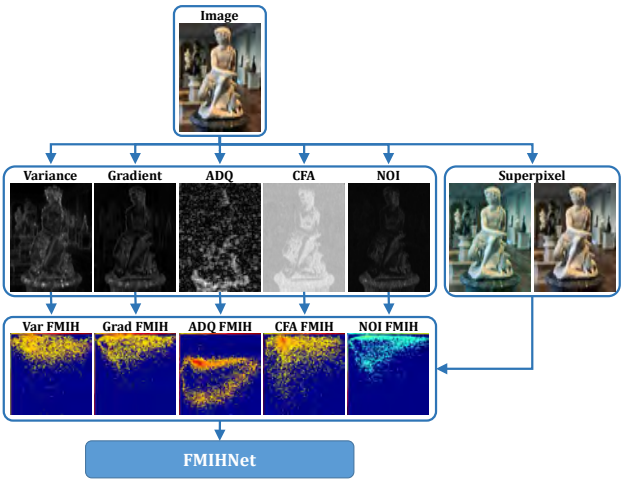


Figure 4. Manipulated refocusing image detection pipeline. The example shown is an iPhone7plus portrait mode image.

are notable differences between the feature maps in these two rows, there is no clear indication of a manipulation except, perhaps, the ADQ feature. And, as mentioned above, the ADQ cue is fragile because it depends on whether blurring is applied after an initial compression.

As mentioned in Sec. 3, the noise cues are signal-dependent in the sense that blurring introduces an inconsistency between intensity and noise levels. To illustrate this, Fig. 3's third row shows scatter plots of the relationship between intensity (on the horizontal axis) and the various features (on the vertical axis). In these plots, particularly the columns related to noise (Variance, Gradient, and NOI), the distinction between the statistics of the authentic image (blue symbols) and the manipulated image (red symbols) becomes quite clear. Noise is reduced in most of the image, though the un-modified foreground region (the red bowl) maintains relatively higher noise because it is not blurred. Note also that the noise levels across the manipulated image are actually *more* consistent than in the authentic image, showing that previous noise-based forensics [20] are ineffective.

5.1. Overall Approach

Fig. 4 shows our forgery detection pipeline, which incorporates the 5 features previously discussed. In order to capture the relationship between individual features and the underlying image intensity, we employ an intensity v.s. feature bivariate histogram – which we call the focus manipulation inconsistency histogram (FMIH). We use FMIH for all five features for defocus forgery image detection, each of which is analyzed by a neural network called FMIHNet. These 5 classification results are combined by a majority voting scheme to determine a final classification label.

First we extract VAR, GRAD, ADQ, CFA and NOI features for each input image, shown in the first five columns of the second row of Fig. 4. Next, we partition the input image

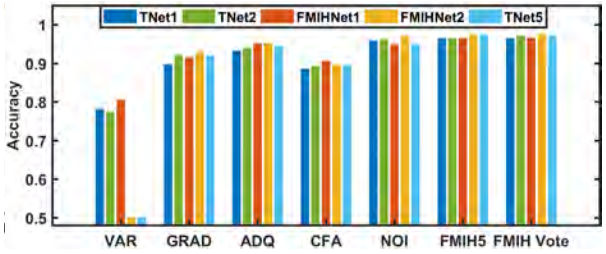


Figure 7. Network architecture analysis. Accuracy changing with different number of layers.

into superpixels and, for each superpixel i_{sp}, we compute the mean $F(i_{sp})$ of each feature measure and its mean intensity. Finally, we generate the FMIH for each of the five figures, shown in the five columns of the third row of Fig. 4. Note that the FMIH are flipped vertically with respect to the scatter plots shown in Fig. 3. An comparison of the FMIH extracted features from the same scene captured with different cameras is shown in Fig. 5.

5.2. Network Architectures

We have designed a FMIHNet, illustrated in Fig. 6, for the 5 histogram features. Our network is a VGG [28] style network consisting of convolutional (CONV) layers with small receptive fields (3×3). During training, the input to our FMIHNet is a fixed-size 101 × 101 FMIH. The FMIHNet is a fusion of two relatively deep sub-networks: FMIHNet1 with 20 CONV layers for VAR and CFA features, and FMIHNet2 with 30 CONV layers for GRAD, ADQ and NOI features. The CONV stride is fixed to 1 pixel; the spatial padding of the input features is set to 24 pixels to preserve the spatial resolution. Spatial pooling is carried out by five max-pooling layers, performed over a 2 × 2 pixel window, with stride 2. A stack of CONV layers are followed by one Fully-Connected (FC) layer, performs 2-way classification. The final layer is the soft-max layer. All hidden layers have the rectification (ReLU) non-linearity.

There are two reasons that we use very small 3 × 3 receptive fields: first, incorporating multiple non-linear rectification layers instead of a single one makes the decision function more discriminative; secondly, this reduces the number of parameters. This can be seen as imposing a regularisation on a lager CONV layer by forcing it to have a decomposition through the 3 × 3 filters.

Because most of the values in our FMIH are zeros (i.e., most cells in the 2D histogram are empty), and because we only have two output classes (authentic and manipulated), more FC layers seem to degrade the training performance. Fig. 7 shows how the accuracy rate changes with different numbers of layers: TNet1 has 11 CONV layers and 3 max-pooling layers, TNet2 has 20 CONV layers and 4 max-pooling layers, TNet3 has 14 CONV layers and 6 max-pooling layers.

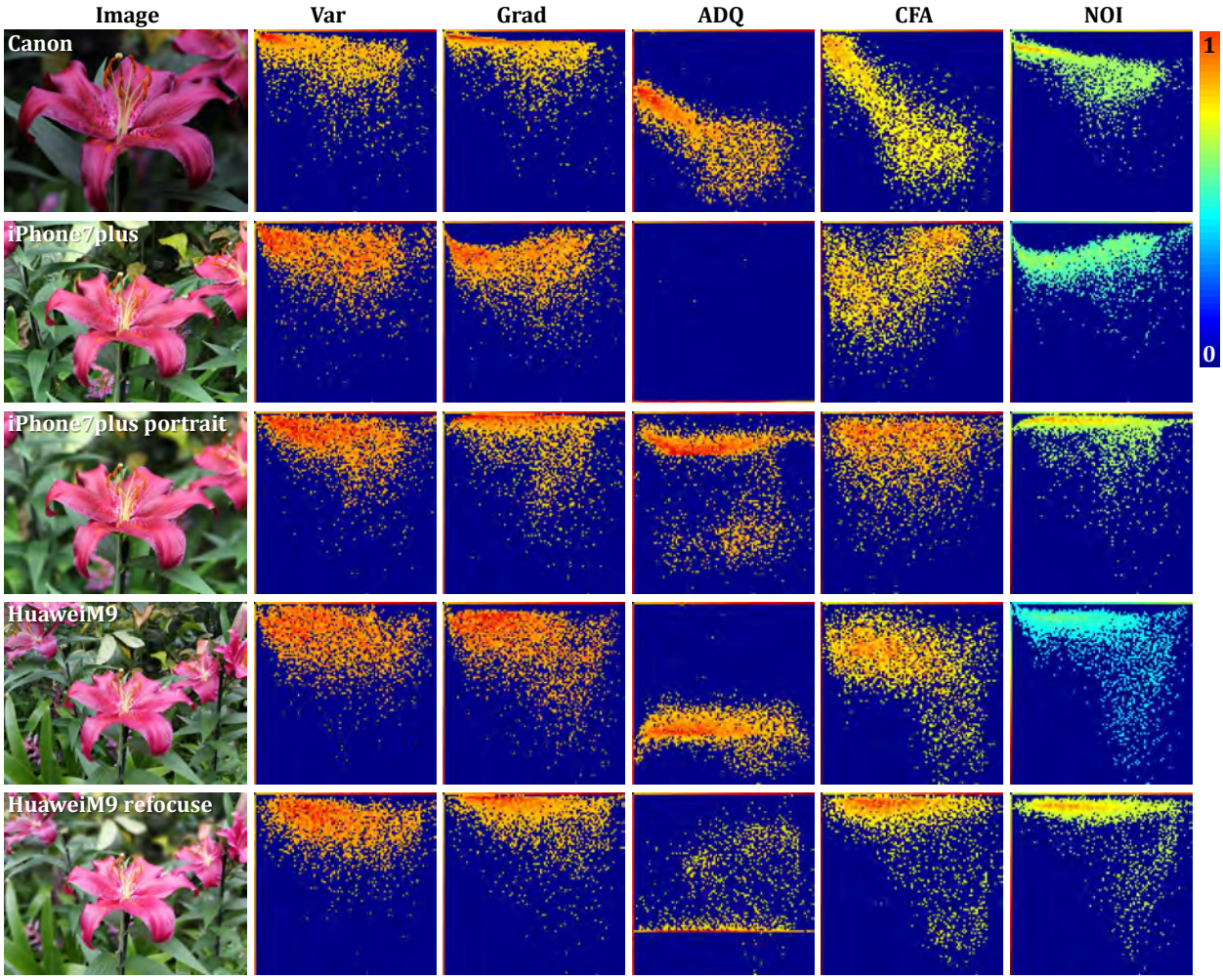


Figure 5. Extracted FMIHs for the 5 feature measures with images captured using Canon60D, iPhone7Plus and HuaweiMate9 cameras.

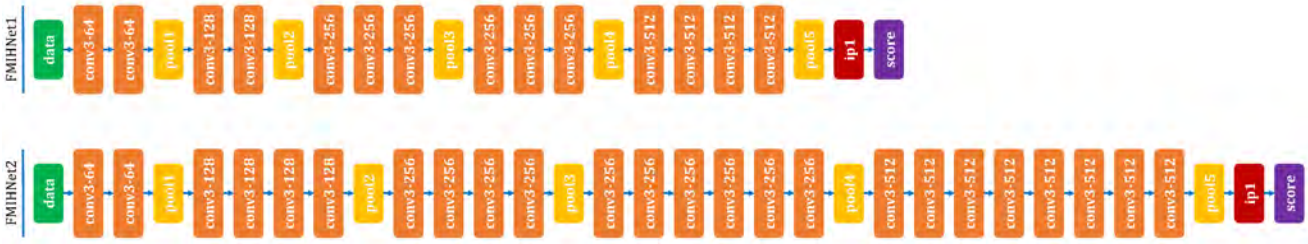


Figure 6. Network architecture: FMIHNet1 for Var and CFA features; FMIHNet2 for Grad, ADQ and NOI features.

6. Experiments

Having introduced a new method to detect focus manipulations, this section will provide quantitative evidence for several claims that we have made throughout the paper. First, we will demonstrate that our method can accurately identify manipulated images *even if they are geometrically correct*. Here, we also show that our method is more accurate than both past forensic methods [11, 13, 20] and the modern vision baseline of CNN classification applied directly to the image pixels. Second, having claimed that the photometric relationship of noise cues with the image intensity is important, we will show that our FMIH histograms are a more useful representation of these cues. Third, we show that our method generalizes from the specific portrait mode manipulations to a wider range of image edits in a standard forensics dataset. Having done so, we can provide a comparison to the method of [12] on that dataset, showing improved performance compared to one of the most recent forensic methods.

6.1. Datasets

To demonstrate our performance on the hard cases of geometrically correct focus manipulations, we have built a focus manipulation dataset (FMD) of images captured with a Canon 60D DSLR and two smartphones having dual lens camera-enabled portrait modes: the iPhone7Plus and the Huawei Mate9. Images from the DSLR represent real shallow DoF images, having been taken with focal lengths in the range 17-70mm and f numbers in the range F/2.8-F/5.6. The iPhone was used to capture aligned pairs of authentic and manipulated images using portrait mode. The Mate9 was also used to capture authentic/manipulated image pairs, but these are only approximately aligned due to its inability to save the image both before and after portrait mode editing.

We use 1320 such images for training and 840 images for testing. The training set consists of 660 authentic images (220 from each of the three cameras) and 660 manipulated images (330 from each of iPhone7Plus and Huawei Mate9). The test set consists of 420 authentic images (140 from each of the three cameras) and 420 manipulated images (140 from each of iPhone7Plus and HuaweiMate9).

In order to test a wider range of manipulations, we also test with the NIST NC2017 [5] dataset. This dataset includes manipulations such as global and local blurring operations, some of which are manually generated and some of which are auto-generated. NC2017 data is more representative of 'in the wild' images, which have variable compression, resizing and source cameras. We sample from this dataset training and testing sets having 1000 images, 500 of which are authentic and 500 of which are manipulated.

6.2. Experiments with FMD

Fig. 5 shows five sample images from FMD, and illustrates the perceptual realism of the manipulated images. Despite this, it's well known that modern deep learning methods can perform quite well given nothing but an image's pixels. The first row of Table. 1 quantifies this performance, and shows that a range of CNN models (AlexNet, CaffeNet, VGG16, VGG19) have classification accuracies in the range of 76-78%. Since our method uses five different feature maps which can easily be interpreted as images, the remaining rows of Table. 1 show the classification accuracies of the same CNN models applied to these feature maps. The accuracies are slightly lower than for the image-based classification.

In Sec. 3, we claimed that a proper accounting for signal-dependent noise via our FMIH histograms improves upon the performance of the underlying features, and this is seen by comparing the image- and feature map-based classification performance of Table. 1 with the FMIH-based classification performanc shown in Table. 2. Using FMIH, even the relatively simple SVMs and LeNet CNNs deliver classification accuracies in the 80-90% range. Our FMIHNet architecture produces significantly better results than these, with our method's voting output having a classification accuracy of 98%. Fig. 8 (a) shows these results in the form of a Receiver Operator Characteristic (ROC) curve, which confirms (1) that the approaches using our FMIH feature representation out-perform the image- or feature map-based approaches and (2) that our network architecture and voting method out-performs other classifiers.

Table 1. Image Classification Accuracy on FMD

Data	AlexNet	CaffeNet	VGG16	VGG19
Image	0.760	0.782	0.771	0.784
VAR map	0.688	0.725	0.714	0.726
GRAD map	0.733	0.767	0.740	0.769
ADQ map	0.735	0.759	0.736	0.740
CFA map	0.761	0.788	0.777	0.785
NOI map	0.707	0.765	0.745	0.760

Table 3. Image Classification Accuracy on NC2017

Data	AlexNet	CaffeNet	VGG16	VGG19
Image	0.497	0.555	0.532	0.567
VAR map	0.483	0.501	0.558	0.504
GRAD map	0.498	0.519	0.504	0.514
ADQ map	0.517	0.502	0.538	0.502
CFA map	0.556	0.559	0.564	0.553
NOI map	0.665	0.609	0.650	0.667

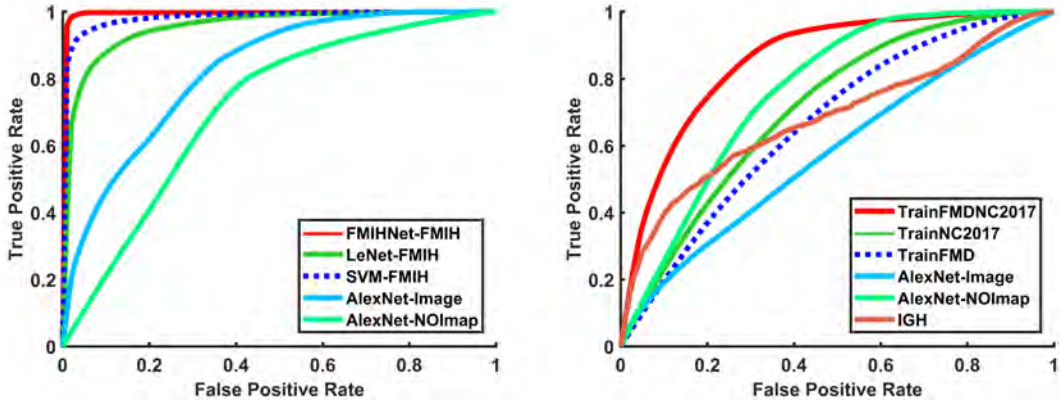
6.3. Experiments with NC2017

As mentioned previously, we train and test with the NC2017 data to represent 'in the wild' images and a wider range of manipulations, not all of which create defocus inconsistencies. Unlike our FMD dataset, the results in Table. 3 show that modern CNN architectures don't perform any better than random when operating on image inputs, and that their accuracy is less than 67% when classifying the various feature maps. Though some of these manipulations lack the visual realism of the portrait mode images, this dataset is in some ways more challenging.

A related challenge is that, while representing a wider range of manipulations, our NC2017 training set has fewer examples. As shown in Fig. 4, the FMIHNet classification accuracy (after voting) is only 63% when trained exclusively on NC2017 data. This is similar to the 62.3% accuracy that we achieve when the network is trained exclusively on the FMD data, which creates a dataset mismatch. Our best NC2017 result, of 77%, is obtained by training our method on the combination of FMD and NC2017 training data. The ROC curves on NC2017 are shown in Fig. 8 (b), where we also include results from [12]'s very recent blur manipulation detector. Our FMIHNet trained on FMD+NC2017 (the red curve) outperforms the other methods on this challenging dataset.

Table 2. FMIH Classification Accuracy on FMD

Data	SVM	AlexNet	CaffeNet	VGG16	VGG19	LeNet	FMIHNet
VAR	0.829	0.480	0.480	0.475	0.512	0.635	**0.850**
GRAD	0.909	0.503	0.500	0.481	0.486	0.846	**0.954**
ADQ	0.909	0.496	0.520	0.503	0.511	0.844	**0.946**
CFA	0.882	0.510	0.520	0.481	0.510	0.871	**0.919**
NOI	0.858	0.497	0.506	0.520	0.530	0.779	**0.967**
Vote	0.942	—	—	—	—	0.888	**0.982**



(a) FMD (b) NC2017

Figure 8. ROC curves for our experiments with (a) our Focus Manipulation Dataset (FMD) and (b) the NIST NC2017 dataset. For both, our method (represented by the red curve) out-performs alternative approaches. The relative ordering of the other curves in (a) demonstrates the benefit of using our FMIH feature representation over using the image or feature maps. The IGH comparison in (b) shows that we out-perform the most recent related work in the area of blur forensics.

Table 4. FMIHNet Classification Accuracy on NC2017

Train on	FMD	NC2017	FMD+NC2017
VAR	0.610	0.612	0.746
GRAD	0.617	0.632	0.673
ADQ	0.546	0.540	0.572
CFA	0.587	0.596	0.685
NOI	0.625	0.615	0.760
Vote	0.623	0.630	**0.771**

7. Conclusion

We have presented a novel framework to detect focus manipulations, which represent an increasingly difficult and important forensics challenge in light of the availability of new camera hardware. Our approach exploits photometric histogram features, with a particular emphasis on noise, whose shapes are altered by the manipulation process. We have adopted a deep learning approach that classifies these 2D histograms separately, and then votes for a final classification. To evaluate this, we have produced a new focus manipulation dataset with images from a Canon60D DSLR, iPhone7Plus, and HuaweiMate9. This dataset includes manipulations, particularly from the iPhone portrait mode, that are geometrically correct due to the use of dual lens capture devices. Despite the challenge of detecting manipulations which are geometrically correct, our method's accuracy is 98%, significantly better than image-based detection with a range of CNNs, and better than prior forensics methods. We also show strong performance on the NIST NC2017 data, which includes a wider range of manipulations over a range of challenging 'in the wild' conditions of compression and other nuisance factors.

Though our new FMD dataset expands the manipulation detection problem beyond simple operations such as splicing and cloning, additional data are needed to cover the range of potential implementations of portrait mode-type features. In particular, additional cameras are needed, as the number of smartphones offering similar features is increasing by the day. We plan to further study the effects of image rescaling and compression on our method, and expand the types of manipulations that we can detect, through continued evaluation of subsequent NIST datasets.

Acknowledgement

This material is based upon work supported by the United States Air Force and the Defense Advanced Research Projects Agency under Contract No. FA8750-16-C-0190. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force or the Defense Advanced Research Projects Agency.

References

[1] Adobe Photoshop. http://www.adobe.com/products/photoshop.html. 1

[2] Google Research Blog. https://research.googleblog.com/2017/10/portrait-mode-on-pixel-2-and-pixel-2-xl.html. 1, 2

[3] iTune Store Depth Effect. https://itunes.apple.com/us/app/depth-effects/id1161218656?mt=8. 1

[4] iTune Store FabFocus. https://itunes.apple.com/us/app/fabfocus-portraits-with-depth-and-bokeh/id1080434313?mt=8. 1

[5] NIST Nimble 2017 Datasets. https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation. 7

[6] K. Bahrami, A. C. Kot, and J. Fan. Splicing detection in out-of-focus blurred images. In *Information Forensics and Security (WIFS), 2013 IEEE International Workshop on*, pages 144–149. IEEE, 2013. 2

[7] K. Bahrami, A. C. Kot, L. Li, and H. Li. Blurred image splicing localization by exposing blur type inconsistency. *IEEE Transactions on Information Forensics and Security*, 10(5):999–1009, 2015. 2

[8] B. A. Barsky. Vision-realistic rendering: simulation of the scanned foveal image from wavefront data of human subjects. In *Proceedings of the 1st Symposium on Applied perception in graphics and visualization*, pages 73–81. ACM, 2004. 2

[9] B. A. Barsky, A. W. Bargteil, D. D. Garcia, and S. A. Klein. Introducing vision-realistic rendering. In *Proc. Eurographics Rendering Workshop*, pages 26–28, 2002. 2

[10] M. Bertalmio, P. Fort, and D. Sanchez-Crespo. Real-time, accurate depth of field using anisotropic diffusion and programmable graphics cards. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pages 767–773. IEEE, 2004. 2

[11] T. Bianchi, A. De Rosa, and A. Piva. Improved dct coefficient analysis for forgery localization in jpeg images. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 2444–2447. IEEE, 2011. 4, 6

[12] C. Chen, S. McCloskey, and J. Yu. Image splicing detection via camera response function analysis. In *Computer Vision and Pattern Recognition, 2017 IEEE Computer Society Conference on*. IEEE, 2017. 2, 6, 7

[13] A. E. Dirik and N. Memon. Image tamper detection based on demosaicing artifacts. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 1497–1500. IEEE, 2009. 3, 4, 6

[14] D. Gillham. Real-time depth-of-field implemented with a postprocessing-only technique. *Shader X5: Advanced Rendering Techniques*, pages 163–175, 2007. 2

[15] J. Göransson and A. Karlsson. Practical post-process depth of field. *GPU Gems*, 3:583–606, 2007. 2

[16] M. Kass, A. Lefohn, and J. Owens. Interactive depth of field using simulated diffusion on a gpu. *Pixar Animation Studios Tech Report*, 2:1–8, 2006. 2

[17] T. J. Kosloff and B. A. Barsky. An algorithm for rendering generalized depth of field effects based on simulated heat diffusion. In *International Conference on Computational Science and Its Applications*, pages 1124–1140. Springer, 2007. 2

[18] M. Kraus and M. Strengert. Depth-of-field rendering by pyramidal image processing. In *Computer Graphics Forum*, volume 26, pages 645–654. Wiley Online Library, 2007. 2

[19] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang. Noise estimation from a single image. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 901–908. IEEE, 2006. 2

[20] B. Mahdian and S. Saic. Using noise inconsistencies for blind image forensics. *Image and Vision Computing*, 27(10):1497–1503, 2009. 3, 4, 5, 6

[21] J. A. Marshall, C. A. Burbeck, D. Ariely, J. P. Rolland, and K. E. Martin. Occlusion edge blur: a cue to relative visual depth. *JOSA A*, 13(4):681–688, 1996. 1

[22] G. Mather. The use of image blur as a depth cue. *Perception*, 26(9):1147–1158, 1997. 1

[23] J. D. Mulder and R. Van Liere. Fast perception-based depth of field rendering. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 129–133. ACM, 2000. 2

[24] M. A. Qureshi and M. Deriche. A bibliography of pixel-based blind image forgery detection techniques. *Signal Processing: Image Communication*, 39:46–74, 2015. 2

[25] G. Riguer, N. Tatarchuk, and J. Isidoro. Real-time depth of field simulation. *ShaderX2: Shader Programming Tips and Tricks with DirectX*, 9:529–556, 2004. 2

[26] P. Rokita. Generating depth of-field effects in virtual reality applications. *IEEE Computer Graphics and Applications*, 16(2):18–21, 1996. 2

[27] C. Scofield. 2 1/2-d depth-of-field simulation for computer animation. In *Graphics Gems III*, pages 36–38. Academic Press Professional, Inc., 1992. 2

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[29] Y. Tsin, V. Ramesh, and T. Kanade. Statistical calibration of ccd imaging process. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 480–487. IEEE, 2001. 2

[30] T. Zhou, J. X. Chen, and M. Pullen. Accurate depth of field simulation in real time. In *Computer Graphics Forum*, volume 26, pages 15–23. Wiley Online Library, 2007. 2