Image Splicing Detection via Camera Response Function Analysis

Can Chen
University of Delaware
Newark, DE, USA
canchen@udel.edu

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

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

Abstract

Recent advances in image manipulation tools have made image forgery detection increasingly more challenging. An important component in such tools is the ability to fake blur to hide splicing and copy-move operations. In this paper, we present a new technique based on the analysis of the camera response functions (CRF) for efficient and robust splicing and copy-move forgery detection and localization. We first analyze how non-linear CRFs affect edges in terms of the intensity-gradient bivariate histograms. We show distinguishable shape differences between real and forged blurs near edges after a splicing operation. Based on our analysis, we introduce a deep-learning framework to detect and localize forged edges. In particular, we show the problem can be transformed to a handwriting recognition problem and resolved by using a convolutional neural network. We generate a large dataset of forged images produced by splicing followed by retouching and comprehensive experiments show our proposed method outperforms the state-of-the-art techniques in accuracy and robustness.

1. Introduction

Forensic algorithm development seeks to improve robustness and expand upon a limited set of cues used to detect media forgery. However, with the advance of image manipulation tools, assessing the integrity of open-source visual media is increasingly difficult. A particularly challenging example is faked motion and/or defocus blurs that are consistent with a plausible scene geometry. Forged blurs can significantly increase photorealism and has shown great success in visual special effects. Since blurs are commonly viewed as a low-pass filter, detecting forged blurs has been particularly challenging and most existing forensic tools lack the scalability and automation needed to provide reliable assessments.

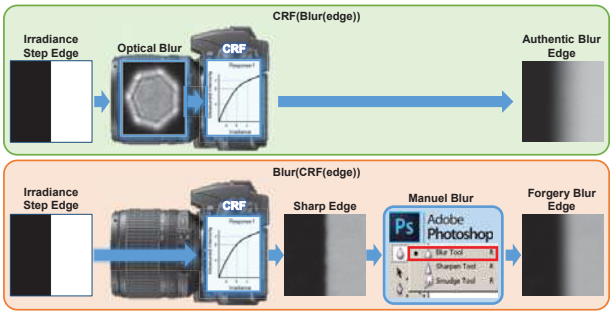

Figure 1. The two paths to a blurred edge. The top row shows the sequence for authentic images, where optical blur happens in the optics before the CRF is applied. The lower row shows the sequence for artificially blurry images, where a sharp edge passes through the CRF and is then blurred in image manipulation software, e.g. PhotoShop.

In this paper, we present a novel forensic based on the Camera Response Function (CRF), which enables a more reliable detection of a wider range of manipulation processes. We specifically target splicing manipulations, where contents are extracted from one image and then copied into a new image. These usually involve a segmentation operation to delimit the content in the original image, the use of which leads to harsh boundaries between it and the background. Because existing forensic methods for splicing detection use properties of different image *regions* for detection, they are prone to false positives when parts of an authentic image have spatially-varying properties. Methods which measure blur consistency between different parts of the image, for instance, are prone to false positives when the different regions are at different depths (and thus have different optical defocus), move separately (and thus have different motion blur), or where there are significant differ-

ences between the underlying textures (since it's not possible to measure blur in a uniformly-colored region). Our method avoids these issues by analyzing *edges*, is able to detect both unnaturally sharp edges (from the segmentation step), and can distinguish naturally-blurred edges from those that have been blurred by image manipulation tools.

Our approach exploits a cue inherent to cameras: the non-linear camera response function (CRF). Recent studies have shown that motion or defocus blurs under CRF are very difficult to compensate [36], because the (linear) blurring operation happens in sequence with the (non-linear) CRF. Traditional image deblurring algorithms can lead to strong ringing artifacts due to CRF non-linearity, if not properly accounted for. At a high level, we exploit this same property for forgery detection. The key idea is that the CRF's non-linearity is not commutative with the linear operation of optical blurring: an edge blurred optically and then passed through the non-linear CRF will have different properties than one which first goes through the CRF and is then blurred (e.g., in Photoshop). Fig.1 shows these two paths, illustrated with a simple step edge. Due to the lack of large volumes of training data, we have found deep learning methods incapable of differentiating the two when operating at the patch level. However, we demonstrate that these two paths lead to distinguishably different statistical relationships between the intensity of a pixel and the magnitude of its gradient, and introduce a deep-learning framework to detect and localize forged edges from this relationship. In particular, we show the problem can be transformed to a handwriting recognition problem and resolved using a convolutional neural network. Our evaluations show improved performance on existing splicing datasets, made entirely of artificially sharp edges, and on a new database of artificially blurred images, where prior methods fail.

2. Related Work

In one of several survey papers on the topic, Farid [10] grouped the different blind image forgery detection approaches under five major categories: (1) pixel-based such as resampling [32, 27], contrast enhancement detection [34] and local descriptors [33, 12]; (2) format-based such as double JPEG compression detection [4]; (3) camera-based such as demosaicing regularity [11, 5, 35], camera response function [15] and sensor pattern noise [6]; (4) physics-based such as light anomalies [20, 17, 22] and size inconsistencies [38]; and (5) geometric-based such as camera intrinsic parameter [19], metric measurement [18], and multi-view geometry [39]. Our technique falls within pixel- and camera-based approaches, among which the most relevant related works check for consistencies between image regions with respect to the CRF and/or blur.

In a photographic camera, as opposed to a machine vision camera, the camera response function (CRF) maps the input irradiance to output image intensity in order to achieve aesthetic objectives. The CRF is unique for each camera, potentially for different capture modes, and can be used as the "fingerprint" of the capturing device, which makes it well suited for image forensics [26, 14, 15]. Most recently, Hsu et al. [15] demonstrated this by automatically segmenting an image into arbitrary-shaped regions, estimating the CRF within each, and then checking the consistency between the segments to determine authenticity. The main issue with this approach is that CRF estimation methods often fail on segments, particularly those with little intensity variation. Because the CRF is a mapping over the full dynamic range of the image, accurate estimation requires the presence of a wide range of intensity values [29], which is often lacking in local image regions. Our method addresses this by using statistical properties of the CRF instead of an explicit estimation of it, and by extracting those properties from edges where blurring necessarily leads to a smoothly-varying range of intensities [36]. Prior methods are also fundamentally unable to detect splicing where the source and target images are captured with the same camera, since both regions will exhibit the same CRF. Other properties can be found in [30] to distinguish photographic images and computer graphics.

Several methods have been proposed to detect forgeries based on inconsistencies between blurring in an image. At a high level, these algorithms check that different regions of the image show the same blur type (motion or defocus) [3], that they have the same extent of blur [2, 37], or that the direction of motion blur is consistent [21]. While these are useful to detect certain types of manipulations, they have a high false alarm rate when presented with authentic images having spatially-varying blur, e.g. from objects at different depths or motion of different velocities. Algorithmically, a key failure point is that the spliced-in regions may not have the strong texture needed to estimate blur, as is the case with many of the images we have captured. These methods also don't count into the effect of the CRF, which have been shown to be important in blur analysis [7, 36].

3. Statistical Traces of Splicing

The nonlinearity of the CRF impacts a lot of image processing and, in particular, its effect is now well-understood on motion debluring [7, 36]. While that work showed how an unknown CRF is a noise source in blur estimation, we demonstrate that it is a key signal helping to distinguish authentic edges from forged splicing boundaries. This allows us to identify forgeries which involve artificially-blurred edges, in addition to ones with artificially-sharp transitions.

We first consider blurred edges: for images captured by a real camera, the blur is applied to scene irradiance as the sensor is exposed, and is then mapped to image intensity via the CRF. There are two operations involved, CRF map-

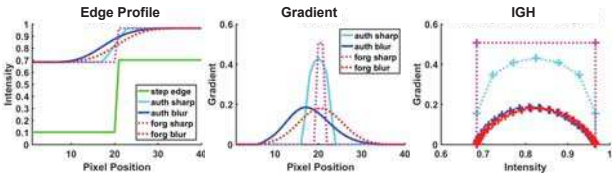

Figure 2. Authentic blur (blue), authentic sharp (cyan), forgery blur (red) and forgery sharp (magenta) edge profiles (left), gradients (center) and IGHs (right).

ping and optical/manual blur convolution. By contrast, in a forged image, the irradiance of a sharp edge is mapped to intensity first then blurred by manual blur point spread function (PSF). The key to distinguishing these is that the CRF is a non-linear operator, so it not commutative, i.e. applying the PSF before the CRF leads to a different result than applying the CRF first, even if the underlying signal is the same.

The key difference is that the profile of an authentic blurred edge (CRF before PSF) is asymmetric w.r.t the center location of edge, whereas the artificially-blurred edge *is* symmetric, as illustrated in Fig.2. This is because, due to its non-linearity, the slope of the CRF is different across the range of intensities. CRFs typically have a larger gradient in low intensities, small gradient in high intensities, and approximately constant slope in the mid-tones. In order to capture this we use the statistical bivariate histogram of pixel intensity vs. gradient magnitude, which we call the intensity gradient histogram (IGH), and use it as the feature for splicing detection. The IGH captures key information about the CRF without having to explicitly estimate it and, as we show later, its shape differs between natural and artificial edge profiles in a way that can be detected with existing CNN architectures.

Before starting the theoretical analysis, we clarify our notation. For simplicity, we study the role of the operations ordering on a 1D edge. We denote the CRF, which is assumed to be a nonlinear monotonically increasing function, as $f(r)$, normalized to satisfy $f(0) = 0$, $f(1) = 1$. And the inverse CRF is denoted as $g(R) = f^{-1}(R)$, satisfying $g(0) = 0$, $f(g(1)) = 1$. r represents irradiance, and R intensity. We assume that the optical blur PSF is a Gaussian function, and have confirmed that the blur tools in Photoshop use a Gaussian kernel,

$$K_g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{x^2}{-2\sigma^2}} \tag{1}$$

When the edge in irradiance space r is a step edge, like the one shown in green in Fig.2

$$H_{step}(x) = \begin{cases} a & x < c \\ b & x \geqq c \end{cases} \tag{2}$$

where x is the pixel location. If we use a unit step function

$$u(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geqq 0 \end{cases} \tag{3}$$

The step edge can be represented by

$$H_{step}(x) = (b-a)u(x-c) + a \tag{4}$$

3.1. Authentically Blurred Edges

An authentically blurred (ab) edge is

$$I_{ab} = f(K_g(x) * H_{step}(x)) \tag{5}$$

where $*$ represents convolution. The gradient of this

$$\begin{aligned} \nabla I_{ab} &= f'(K_g(x) * H_{step}(x)) \cdot \frac{d[K_g(x) * H_{step}(x)]}{dx} \\ &= f'(K_g(x) * H_{step}(x)) \cdot K_g(x) * \frac{dH_{step}(x)}{dx} \end{aligned}$$

Because the differential of the step edge is a delta function,

$$\frac{dH_{step}(x)}{dx} = (b-a)\delta(x-c) \tag{6}$$

We have

$$\begin{aligned} \nabla I_{ab} &= f'(K_g(x) * H_{step}(x)) \cdot K_g(x) * (b-a)\delta(x-c) \\ &= (b-a)f'(K_g(x) * H_{step}(x)) \cdot K_g(x-c) \end{aligned}$$

Substituting 5 into above equation, we have the relationship between I_{ab} and gradient ∇I_{ab}

$$\nabla I_{ab} = (b-a)f'(f^{-1}(I_{ab})) \cdot K_g(x-c) \tag{7}$$

And $K_g(x-c)$ is just shifting the blur kernel to the location of the step. Because f is nonlinear and its gradient is large at lower irradiance and small at higher irradiance, f' is asymmetric. Therefore IGH of authentically blurred edge is asymmetric, as shown in blue in Fig.2.

3.2. Authentic Sharp Edge

In real images, the assumption that a sharp edge is a step function does not hold. Some [31] assume a sigmoid function, for simplicity, whereas we choose a small Gaussian kernel to approximate the authentic sharp (as) edge.

$$I_{as} = f(K_s(x) * H_{step}(x)) \tag{8}$$

The IGH is the same as that of an authentic blur edge 7, with the size and σ of the kernel being smaller. Because the blur extent is very small, there is a small transition edge region, and the effect of CRF will not be very obvious. The IGH remains symmetric shown as cyan in Fig.2.

3.3. Forged Blurred Edge

The model for a forged blurred (fb) edge is

$$I_{fb} = K_g(x) * f(H_{step}(x))$$

The gradient of this is

$$\begin{aligned}\nabla I_{fb} &= \frac{d[K_g(x) * f(H_{step}(x))]}{dx} \\ &= K_g(x) * [f'(H_{step}(x)) \cdot H'_{step}(x)]\end{aligned}$$

Because

$$f'(H_{step}(x)) = (f'(b) - f'(a))u(x-c) + f'(a) \quad (9)$$

and

$$f(x) \cdot \delta(x) = f(0) \cdot \delta(x), \quad (10)$$

we have

$$\begin{aligned}\nabla I_{fb} &= K_g(x) * [f'(b)\delta(x-c)] \\ &= (b-a)f'(b)K_g(x) * \delta(x-c) \\ &= (b-a)f'(b)K_g(x-c).\end{aligned}$$

Clearly ∇I_{fb} has the shape the same as the PSF kernel, which is symmetric w.r.t. the location of the step c, shown as red in Fig.2.

3.4. Forgery Sharp Edge

A forged sharp (fs) edge appears as an unnaturally abrupt jump in intensity as

$$I_{fs} = f(H_{step}(x)) \quad (11)$$

The gradient of the spliced sharp boundary image is

$$\begin{aligned}\nabla I_{fs} &= \frac{dI_{fs}}{dx} = \frac{d[f(H_{step}(x))]}{dx} \\ &= f'(H_{step}(x)) \cdot (b-a)\delta(x-c) \\ &= f'(b) \cdot \delta(x-c)\end{aligned}$$

There are only two intensities a and b, both having the same (large) gradient. The IGH of a forged sharp edge will only have all pixels fall in only two bins shown as magenta in Fig.2.

3.5. Distinguishing IGHs of the Edge Types

To validate the theoretical analysis, we show IGHs of the 4 types of edge from real images in Fig.3. An authentically-blurred edge is blurred first, inducing intensity values between the low and high values in irradiance. This symmetric blur profile then is mapped through the CRF, becoming asymmetric.

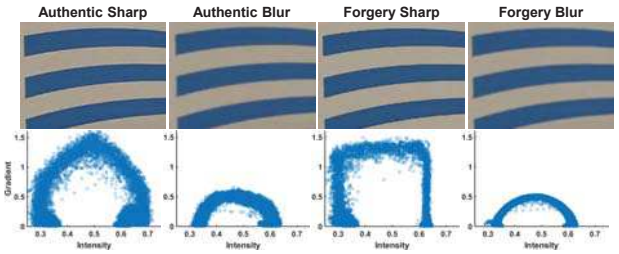

Figure 3. The figure shows the 4 different edge classes and their IGH. Notice that the IGHs have different scales.

In the forged blur edge case, by contrast, the irradiance step edge is mapped by CRF first, to a step edge in intensity space. Then the artificial blur (via PhotoShop, etc.) induces values between the two intensity extrema, and the profile reflects the symmetric shape of PSF.

The forgery sharp edge is an ideal step edge, whose nominal IGH has only two bins with non-zero counts. However, due the existence of noise in images captured by cameras, shading, etc., the IGH of forgery sharp edge appears a rectangular shape as shown in Fig.3. The horizontal line shows that the pixels fall into bins of different intensities with the same gradient, is caused by the pixel intensity varying along the sharp edge. The two vertical lines shows that the pixels fall into bins of different gradient with similar intensity values, is caused by the pixels intensity varying on the constant color regions.

As for the authentic sharp edge, the IGH is easily confused with a forged blurred edge, in that both are symmetric. If we only consider the shape of IGH, this would lead to a large number of false alarms. To disambiguate the two, we add an additional feature: the absolute value of the center intensity and gradient for each bin. This value helps due to the fact that at the same intensity value, the gradient of blurred edge is always smaller than the sharp edge. With our IGH feature, we are able to detect splicing boundaries that are hard for prior methods, such as spliced regions having constant color or those captured by the same camera.

4. Classifying IGHs

Having described the IGH and how these features differ between the four categories of edges, we now consider mechanisms to solve the inverse problem of inferring the edge category from an image patch containing an edge. Since our IGH is similar to bag of words-type features used extensively in computer vision, SVMs are a natural classification mechanism to consider. In light of the recent success of powerful Convolutional Neural Networks (CNNs) applied directly to pixel arrays, we consider this approach, but find that the lack of training data limits its performance. To address this, we map the classification problem from the data-starved patch domain to a character shape recognition in the IGH domain, for which we leverage existing CNN

architectures and training data.

4.1. SVM Classification

As in other vision applications, SVMs with the histogram intersection kernel [28] performs best for IGH classification. We unwarp our 2D IGH into a vector, and append the center intensity and gradient value of each bin. Following the example of SVM-based classification on bag of words features, we use a multi-classification scheme by training 4 one-vs-all models: authentic vs others, authentic sharp vs. others, forged sharp vs others, and forged blur vs others. Then we combine the scores of all 4 models to classify the IGH as either authentic or a forgery.

4.2. CNN on Edge Patches

CNNs have revolutionized a wide range of computer vision areas, showing that they can out-perform approaches with hand-crafted features, so we evaluate whether a CNN applied directly to edge patches can out-perform methods involving our IGH. We use the very popular Caffe [16] implementation for the classification task.

We first train a CNN model on edge patches. In order to examine the difference between authentic and forged patches, we synthesized the authentic and forged blur processes on white, gray and black edges as shown in Fig.4. Given the same irradiance map, we apply the same Gaussian blur and CRF mapping in two orderings shown in Fig. 1. The white region in the authentically blurred image always appears to be larger than in forgery blur image, because the CRF has higher slope in the low intensity region. That means all intensity of an authentically blurred edge is brought to the white point faster than a forgery. This effect can also be observed in real images, such as the Skype logo in Fig.4, where the white letters in the real image are bolder than in the forgery. Another reason for this effect is that cameras have limited dynamic range. A forged sharp edge will appear like the irradiance map with step edges, while an authentically sharp edge would be easily confused with forgery blur since this CRF effect is only distinguishable with a relatively large amount of blur. Thus the transition region around the edge *potentially* contains a cue for splicing detection in a CNN framework.

4.2.1 CNN on IGHs

Our final approach marries our IGH feature with the power of CNNs. Usually, people wouldn't use a histogram with CNN classification because spatial arrangements are important for other tasks, e.g. object recognition. But, as our analysis has shown, the cues relevant to our problem are found at the pixel level, and our IGH can be used to eliminate various nuisance factors: the orientation of the edge, the height of the step, etc. This has an advantage of reducing the training data dimensionality, and thus the large number of training data needed to produce accurate models. Lacking an ImageNet-scale training set for forgeries, this is a key advantage of our method.

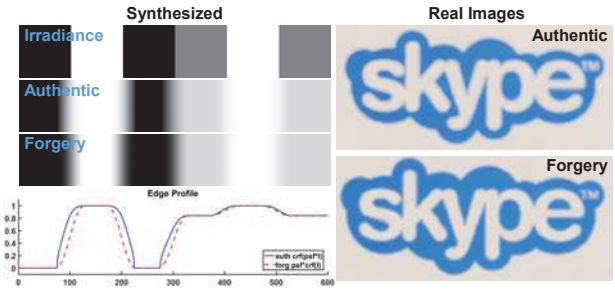

Figure 4. Authentic vs forgery images and edge profiles. The synthesized images are using the same CRF and Gaussian kernel. The real images are captured by Canon 70D. The forgery image is generated by blurring a sharp image the same amount to match the authentic image.

In the sense that a quantized IGH looks like various U-shaped characters, our approach reduces the problem of edge classification into a handwritten character recognition problem, which has been well studied [23]. Since we only interested in the shape of the IGH, the LeNet model [23] is very useful for the IGH recognition task.

5. Automating IGH-based Forensics

In order to automate the forensic analysis of images, our method requires that image patches be identified, from which the IGH features are extracted. After each such patch/feature is classified as authentic or forged, we combine these local results to generate a mask indicating regions of the image we believe to have been spliced in, to facilitate comparisons with prior work. The overall pipeline is shown in Fig. 5, and steps are described below.

5.1. IGH Feature Extraction

Given an image I, we extract edge patches of size $n \times n$ pixels ($n \in [20, 200]$), which are used for direct CNN classification and IGH extraction. First, we clarify our notation. We denote by the vector $\mathbf{C}_L = [C_{L,1}, C_{L,2}, \cdots]$ all pixels with lower intensities, the vector $\mathbf{C}_H = [C_{H,1}, C_{H,2}, \cdots]$ all pixels with higher intensities and $\mathbf{C}_E = [C_{E,1}, C_{E,2}, \cdots]$ all pixels on edges. We require that each patch have only two colors $\mathbf{C}_L$ and $\mathbf{C}_H$, in addition to the transition region. Selected patches satisfy the following conditions:

Color Variance Check The two color regions need to be constant.

$$variance(\mathbf{C}_L) < threshold_{cv}$$
$$variance(\mathbf{C}_H) < threshold_{cv}$$

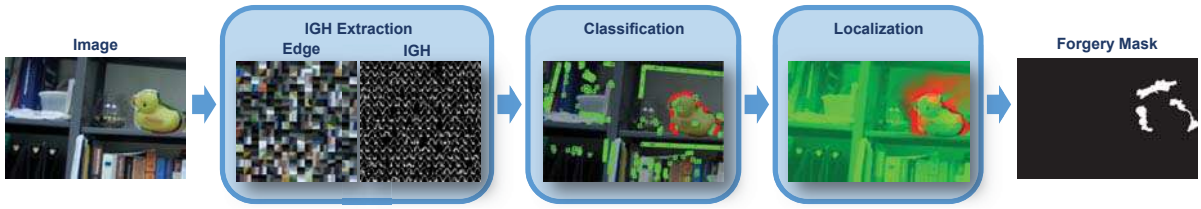

Figure 5. Our automated splicing detection pipeline, which selects edge patches, computes IGHs, applies local classification, and then localizes the spliced-in region to create a binary forgery mask.

Color Difference Check We discard edges with a low contrast because the ratio between the edge profile and noise would be too low.

$$|mean(\mathbf{C}_H) - mean(\mathbf{C}_L)| < threshold_{cd}$$

Area Difference Check We need enough size of $\mathbf{C}_L$ and $\mathbf{C}_H$ for statistic analysis.

$$||\mathbf{C}_H|| - ||\mathbf{C}_L|| < 0.5 * (||\mathbf{C}_H|| + ||\mathbf{C}_L||)$$

Edge Range Check The colors along the edge are a mixture of $\mathbf{C}_L$ and $\mathbf{C}_H$.

$$max(\mathbf{C}_E) < max(\mathbf{C}_H)$$
$$min(\mathbf{C}_E) < min(\mathbf{C}_L)$$

For valid each patch that satisfies all the check conditions, we compute the gradient and finally the IGH for classification.

In real life photography, people like to use the out-of-focus blur for aesthetic purposes. Therefore, they could splice object into the defocused region. In order to detect the blur edges, previous work like [8, 25] filtered edge candidates by keeping only high contrast isolated straight-line edges, and can only observe one edge at a time. We construct a very simple yet efficient scheme to deal with blur edges. We first downsample the image I to $1/5$ as I_{down}, then apply the canny edge detector on I_{down}, giving $edge_{down}$. Finally, we upsample $edge_{down}$ to the original size of I. Now we have obtained the edge region that large enough to cover the whole transition between $\mathbf{C}_H$ and $\mathbf{C}_L$.

5.2. Handling Large Blur

In case of large amounts of blur, we adopt a multi-scale scheme by first downsampling the edge patch to $1/4$, and then compute the IGH on the downsampled patch. The effect is shown in Fig.6.

When the blur amount is large, the kernel size may be larger than some of the structures in the image, such as a printed character, thin lines or, if there are repeated patterns of these thin structures, the convolution will integrate more than one of these edges, introducing points inside the arch shape. Downsampling the image eliminates these inside pixels, bringing back the original shape.

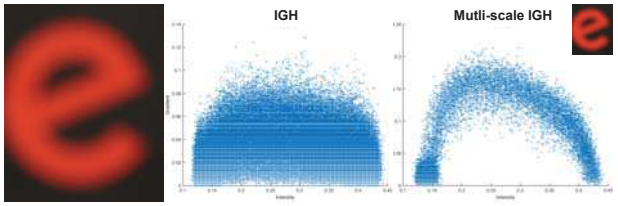

Figure 6. Original size and downsampled edge patch and IGH. IGHs are in different scales.

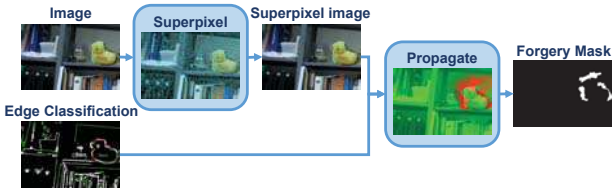

Figure 7. The mask generation step, in which we propagate edge classification results in the superpixel image. In edge classification, red is forgery, green is authentic, white refers to edges not selected.

5.3. Mask Generation

The goal of this step is to generate, from point-wise detections of forged edges, a binary mask indicating the spliced-in region. We first segment the image using superpixel and set the color of each pixel in the output image to the mean RGB color of the superpixel region. We then propagate the edge classification results on the superpixel image using colorization [24], with red representing forged and green authentic superpixels. A threshold of 50% red is set to generate the final forgery mask. The process is shown in Fig.7.

6. Splicing Logo Dataset

Among the splicing datasets made available [1, 9, 13], only the Columbia Uncompressed Image Splicing Dataset (CUISDE) [13], which has 183 authentic and 180 spliced images, includes the masks needed for training. However, none of the existing datasets include forgeries where the splicing boundary is blurred to reduce the harshness of the segmentation, despite this being a natural way to hide the forgery. To address this, we build our own Splicing Logo Dataset (SpLogo) containing 1533 authentic images and 1277 forged blur images of logos with different colors and different amounts of blur. All the images are taken by a

Canon70D. The optical blur amount is manipulated via two different settings: one is by changing the focal plane (lens moving), the other is by changing the aperture with focal plane slightly off the logo plane. The logos are printed on a sheet of paper, and the focal plane is set to be parallel to the logo plane to eliminate the effect of depth related defocus. The forgery images are generated to match the amount of optical blur through a optimization routine. As shown in Fig. 8, the forged and authentic blurred logos in SpLogo are hard to tell apart, unlike CUISDE forgeries which look very artificial (see Fig. 11).

7. Experiments

We test our CNN and SVM methods on SpLogo, CUISDE datasets and a combination of the two. The patch size is $max(min(img_height, img_width)/25, 12)$. $threshold_{cv} \in [10/255, 30/255]$ and $threshold_{cd} \in [10/255, 20/255]$. We use a histogram intersection kernel [28] for our SVM, LeNet [23] for CNN with a $1e^{-6}$ base learning rate, $1e^{6}$ maximum number of iterations. The training set contains 1000 authentic and 1000 forged patches.

Table.1 compares the accuracy of the different approaches described in Sec. 4. Somewhat surprisingly, the CNN applied directly to patches does the worst job of classification. Among the classification methods employing our hand-crafted feature, CNN classifiers obtain better results than SVM, and adding the absolute value of intensity and gradient to IGH increases classification accuracy.

7.1. Experiment on SpLogo Dataset

To test our ability to detect forged blurs, we generate some concatenated images from our SpLogo dataset containing authentic sharp, authentic blur and forged blur subimages. We first test the performance of IGH w/o values (IGHnoV) vs. IGH. The results are shown in Fig.8, with green patches representing those classified as authentic, red as forgery. The IGH performs better than IGHnoV using both SVM and CNN, eliminating false positives of authentic sharp edges being identified as forgeries.

We also tested the effect of multi-scale scheme to deal with large amount of blur. In this test, we further test the classification of authentic sharp (dark green), authentic blur (bright green), forgery sharp (bright red) and forgery blur (dark red). Notice that in Fig.9, some of the edges on left authentic sharp 'e' are misclassified as forgery sharp and authentic blur, and middle authentic blur 'e' misclassified as forgery. Using the multi-scale scheme, all authentic edges are correctly classified, except the SVM cannot clearly differentiate authentic sharp and authentic blur, while the CNN performs better even with authentic sharp and blur.

We include all four types of edges to test the different classification methods. CNN with IGHV performs the best of the three methods.

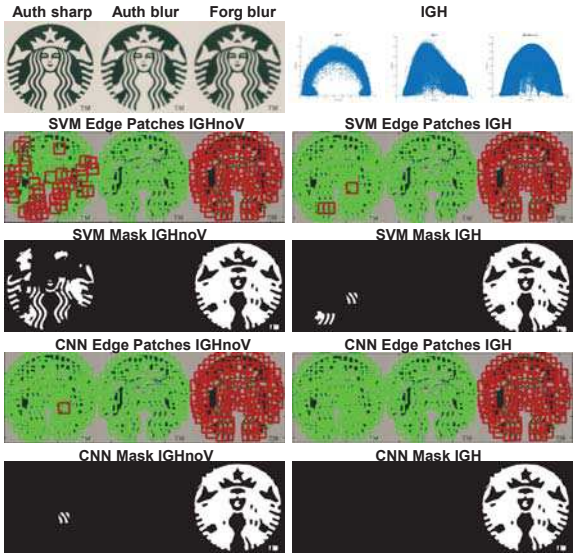

Figure 8. IGH performance test. Green patches are classified as authentic, and red are classified forgeries. First column is the IGHnoV classification result, and second column is IGH. IGHs are in different scales.

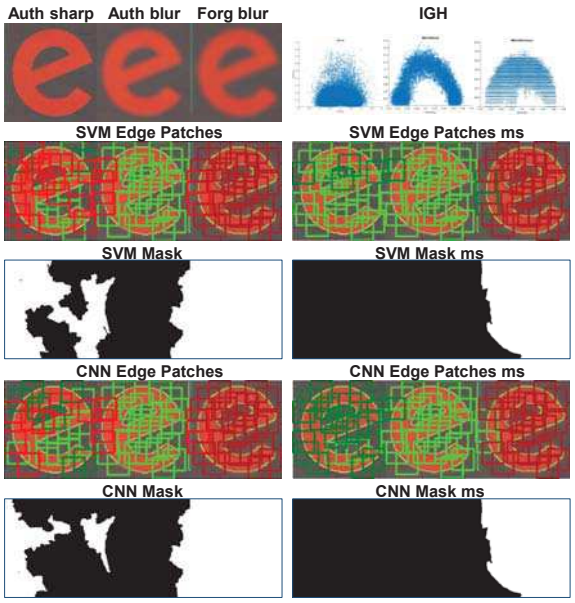

Figure 9. Multi-scale effect. First column is the classification result without multi-scale, and second column is with multi-scale. IGHs are in different scales. **Best viewed on a display**

Table 2. Patch Accuracy on CUISDE.

Classifier	Precision	Recall	Accuracy
CRF SVM [15, 13]	0.7	0.7	0.87
IGH SVM	0.93	0.94	0.94
IGH CNN	**0.95**	**0.95**	**0.97**

7.2. Experiment on CUISDE Dataset

The Columbia Splicing dataset has a lot of randomly chosen splicing regions, as shown in the 3rd and 5th columns of Fig.11. However, it is useful because it was collected using a number of different cameras with differ-

Table 1. Patch Classification Accuracy.

Classifier	Image			IGHnoV			IGH		
	CUISDE	SpLogo	Combine	CUISDE	SpLogo	Combine	CUISDE	SpLogo	Combine
SVM	-	-	-	0.924	0.972	0.937	0.940	0.972	0.951
CNN	0.888	0.896	0.891	0.943	0.972	0.979	**0.97**	**0.99**	**0.978**

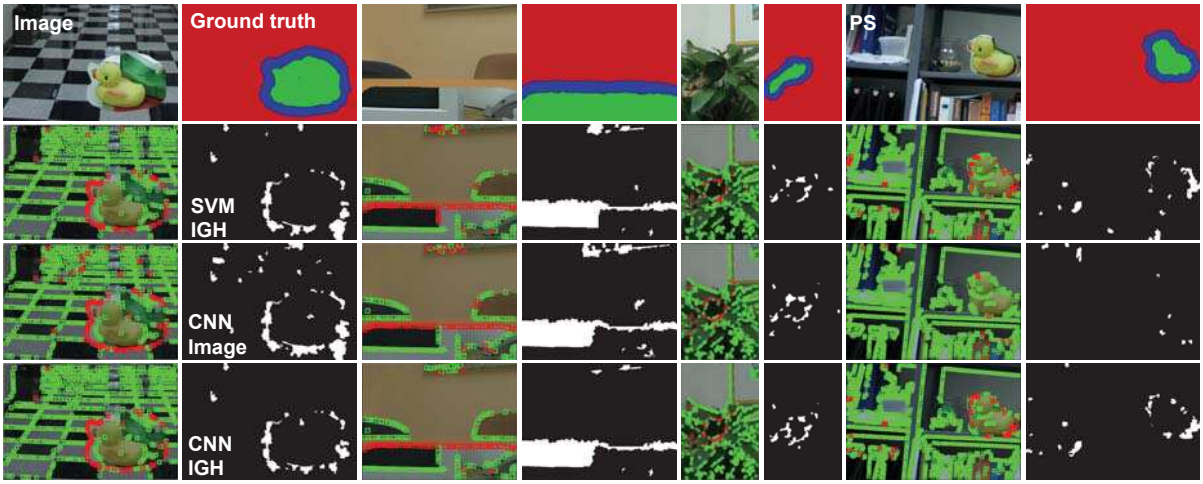

Figure 11. Results on CUISDE. In the ground truth image, blue is the forged edge, and red and green segments come from different images. The two columns show results of a PSed splice boundary image. Our results show binary masks of spliced-in regions.

Figure 10. SVM and CNN results on SpLogo dataset. Both SVM classification and CNNs applied to the patches give false negatives on forged sharp edges, which are reduced significantly by applying the CNN to our histogram feature.

ent CRFs, compensating for SpLogo's dependance on a single camera. In our results, some of the splicing boundaries in CUISDE aren't selected using our edge patch selection scheme. And since our propagation method depends on colors, we can only obtain the edge region not a full mask of the spliced region. Our CNN model trained on IGH performs the best, reducing the false alarm rate compare to the other two models. Our algorithm also outperforms the prior art [15, 13] in precision and recall, as shown in Table.2.

8. Conclusion

We present a novel method to detect spliced-in regions of a forged image, with complete automation in patch selection and mask generation. To demonstrate that our method can handle spliced-in regions having nearly constant intensity, which fools prior methods, we have captured a new SpLogo dataset on which we demonstrate excellent performance. This is in addition to our state-of-the-art performance on the existing CUISDE dataset, which contains only artificially sharp edges. The key to our detection method is that a non-linear CRF leads to differences in pixel-level image statistics depending on whether it is applied before or after blurring. Our IGH feature captures these statistics, and provides a way to eliminate nuisance variables such as edge orientation and step height so that CNN methods can be applied despite a lack of a large training set.

Though SpLogo expands the splice detection problem beyond artificial-looking sharp boundaries, additional data are needed to address the full range of potential manipulations. In particular, additional cameras (with different CRFs) are needed, as are more complicated scenes. As well, we plan to study the effects of image compression on our method, since many forensics are known to be sensitive to image quality and the presence of blocking artifacts.

Acknowledgement

All brand names and trademarks used herein are for descriptive purposes only and are the property of their respective owners. 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] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, L. Del Tongo, and G. Serra. Copy-move forgery detection and localization by means of robust clustering with j-linkage. *Signal Processing: Image Communication*, 28(6):659–669, 2013. 6

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

[3] 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

[4] T. Bianchi and A. Piva. Image forgery localization via block-grained analysis of jpeg artifacts. *IEEE Transactions on Information Forensics and Security*, 7(3):1003–1017, 2012. 2

[5] H. Cao and A. C. Kot. Accurate detection of demosaicing regularity for digital image forensics. *IEEE Transactions on Information Forensics and Security*, 4(4):899–910, 2009. 2

[6] M. Chen, J. Fridrich, M. Goljan, and J. Lukás. Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security*, 3(1):74–90, 2008. 2

[7] X. Chen, F. Li, J. Yang, and J. Yu. A theoretical analysis of camera response functions in image deblurring. In *European Conference on Computer Vision*, pages 333–346. Springer, 2012. 2

[8] T. S. Cho, S. Paris, B. K. Horn, and W. T. Freeman. Blur kernel estimation using the radon transform. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 241–248. IEEE, 2011. 6

[9] J. Dong, W. Wang, and T. Tan. Casia image tampering detection evaluation database. In *Signal and Information Processing (ChinaSIP), 2013 IEEE China Summit & International Conference on*, pages 422–426. IEEE, 2013. 6

[10] H. Farid. Image forgery detection–a survey. 2009. 2

[11] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva. Image forgery localization via fine-grained analysis of cfa artifacts. *IEEE Transactions on Information Forensics and Security*, 7(5):1566–1577, 2012. 2

[12] J. Fridrich and J. Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012. 2

[13] Y.-F. Hsu and S.-F. Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *2006 IEEE International Conference on Multimedia and Expo*, pages 549–552. IEEE, 2006. 6, 7, 8

[14] Y.-F. Hsu and S.-F. Chang. Image splicing detection using camera response function consistency and automatic segmentation. In *2007 IEEE International Conference on Multimedia and Expo*, pages 28–31. IEEE, 2007. 2

[15] Y.-F. Hsu and S.-F. Chang. Camera response functions for image forensics: an automatic algorithm for splicing detection. *IEEE Transactions on Information Forensics and Security*, 5(4):816–825, 2010. 2, 7, 8

[16] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 5

[17] M. K. Johnson and H. Farid. Exposing digital forgeries by detecting inconsistencies in lighting. In *Proceedings of the 7th workshop on Multimedia and security*, pages 1–10. ACM, 2005. 2

[18] M. K. Johnson and H. Farid. Metric measurements on a plane from a single image. *Dept. Comput. Sci., Dartmouth College, Tech. Rep. TR2006-579*, 2006. 2

[19] M. K. Johnson and H. Farid. Detecting photographic composites of people. In *International Workshop on Digital Watermarking*, pages 19–33. Springer, 2007. 2

[20] M. K. Johnson and H. Farid. Exposing digital forgeries in complex lighting environments. *IEEE Transactions on Information Forensics and Security*, 2(3):450–461, 2007. 2

[21] P. Kakar, N. Sudha, and W. Ser. Exposing digital image forgeries by detecting discrepancies in motion blur. *IEEE Transactions on Multimedia*, 13(3):443–452, 2011. 2

[22] E. Kee and H. Farid. Exposing digital forgeries from 3-d lighting environments. In *2010 IEEE International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2010. 2

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5, 7

[24] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 689–694. ACM, 2004. 6

[25] S. Lin, J. Gu, S. Yamazaki, and H.-Y. Shum. Radiometric calibration from a single image. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–938. IEEE, 2004. 6

[26] Z. Lin, R. Wang, X. Tang, and H.-Y. Shum. Detecting doctored images using camera response normality and consistency. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 1087–1092. IEEE, 2005. 2

[27] B. Mahdian and S. Saic. Blind authentication using periodic properties of interpolation. *IEEE Transactions on Information Forensics and Security*, 3(3):529–538, 2008. 2

[28] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 5, 7

[29] Y. Matsushita and S. Lin. Radiometric calibration from noise distributions. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 2

[30] T.-T. Ng, S.-F. Chang, J. Hsu, L. Xie, and M.-P. Tsui. Physics-motivated features for distinguishing photographic images and computer graphics. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 239–248. ACM, 2005. 2

[31] T.-T. Ng, S.-F. Chang, and M.-P. Tsui. Using geometry invariants for camera response function estimation. In *2007*

IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2007. 3

[32] A. C. Popescu and H. Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on signal processing*, 53(2):758–767, 2005. 2

[33] Y. Q. Shi, C. Chen, G. Xuan, and W. Su. Steganalysis versus splicing detection. In *International Workshop on Digital Watermarking*, pages 158–172. Springer, 2007. 2

[34] M. C. Stamm and K. R. Liu. Forensic detection of image manipulation using statistical intrinsic fingerprints. *IEEE Transactions on Information Forensics and Security*, 5(3):492–506, 2010. 2

[35] A. Swaminathan, M. Wu, and K. R. Liu. Digital image forensics via intrinsic fingerprints. *IEEE Transactions on Information Forensics and Security*, 3(1):101–117, 2008. 2

[36] Y.-W. Tai, X. Chen, S. Kim, S. J. Kim, F. Li, J. Yang, J. Yu, Y. Matsushita, and M. S. Brown. Nonlinear camera response functions and image deblurring: Theoretical analysis and practice. *IEEE transactions on pattern analysis and machine intelligence*, 35(10):2498–2512, 2013. 2

[37] J. Wang, G. Liu, B. Xu, H. Li, Y. Dai, and Z. Wang. Image forgery forensics based on manual blurred edge detection. In *2010 International Conference on Multimedia Information Networking and Security*, pages 907–911. IEEE, 2010. 2

[38] H. Yao, S. Wang, Y. Zhao, and X. Zhang. Detecting image forgery using perspective constraints. *IEEE Signal Processing Letters*, 19(3):123–126, 2012. 2

[39] W. Zhang, X. Cao, Z. Feng, J. Zhang, and P. Wang. Detecting photographic composites using two-view geometrical constraints. In *ICME*, pages 1078–1081, 2009. 2