applied optics

Focal stack based image forgery localization

Zhengyu Huang, Jeffrey A. Fessler, and Theodore B. Norris*

Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, Michigan, 48105, USA
*Corresponding author: tnorris@umich.edu

Received 6 December 2021; revised 16 April 2022; accepted 18 April 2022; posted 19 April 2022; published 2 May 2022

Image security is becoming an increasingly important issue due to advances in deep learning based image manipulations, such as deep image inpainting and deepfakes. There has been considerable work to date on detecting such image manipulations using improved algorithms, with little attention paid to the possible role that hardware advances may have for improving security. We propose to use a focal stack camera as a novel secure imaging device, to the best of our knowledge, that facilitates localizing modified regions in manipulated images. We show that applying convolutional neural network detection methods to focal stack images achieves significantly better detection accuracy compared to single image based forgery detection. This work demonstrates that focal stack images could be used as a novel secure image file format and opens up a new direction for secure imaging.

https://doi.org/10.1364/AO.450654

1. INTRODUCTION

Digital images are convenient to store and share, but they are also susceptible to malicious manipulations. With common photo editing tools, little effort or expertise is needed to convincingly manipulate an image. With advances in deep learning, this issue becomes even more severe: generative adverserial networks (GANs) are able to synthesize realistic non-existing images, change the style of an image, or inpaint an image to remove specific objects in it. Deepfakes can even seamlessly swap the face of one person with another in images [1,2]. These maliciously manipulated images could appear in the news, causing misleading opinions in the public or being provided in the court as evidence, with obvious serious consequences.

Verifying the integrity of multi-media has been a research topic for a long time in the field of multi-media forensics [3–8]. Traditional methods verify the integrity of a digital medium and detect traces of malicious manipulation by examining some signatures in the image, using either passive or active approaches. In the active approach, semi-fragile watermarks are pro-actively embedded into the image. The introduced watermark (which is visually imperceptible) is persistent after benign image operations such as brightness adjustment, resizing, and compression, but can be destroyed by malicious editing. In the passive approach, imaging artifacts such as those due to lens distortion [9], color filtering [7], photo response non-uniformity (PRNU) [8], or compression are used to authenticate an image.

Each method has its own limitations, however. The passive approach, while being simple to implement, relies on weak traces that are likely to be destroyed by compression/resizing. PRNU fingerprint analysis, while being a popular forensic method, requires knowledge about the source camera's PRNU. On the other hand, the active watermarking approach is more robust against compression/resizing, but alters the original content due to the watermark embedding. More recently, deep learning based forensic detection methods have also been proposed [10–13]. However, the ability to generalize data-driven models remains a key challenge: these models perform well on images that are similar to the training data, but the performance can quickly degrade when the models are fed with images that differ too much from the training data distribution [14,15].

Most existing image forgery detection methods assume a standard conventional camera and attempt to determine image authenticity by analyzing features present in a given 2D image file. Such methods are widely applicable to present-day 2D image file formats, but forgery detection remains a significant and growing problem as the sophistication of image manipulation techniques continues to grow.

In this paper, we propose to make image manipulation and forgery more detectable through a combined hardware and software approach. Specifically, we propose to use a *focal stack* of images, instead of a single image, for secure media sharing, where the entire focal stack image file is shared publicly. By enriching the information carried by the digital images, essentially extending the data into a third dimension, we can dramatically improve our ability to detect image manipulation. Because the image formation requires a focal stack, this approach may not be as widely applicable as present-day 2D imaging approaches; it is nevertheless critical to consider alternatives that may involve more complex optical systems for imaging where security is an over-riding concern, given the severe limitations faced by 2D image forgery detection.

Figure 1 illustrates the idea: images in the focal stack contain depth-dependent defocus blur. Because generating physically realistic content with defocus blur that is consistent across the

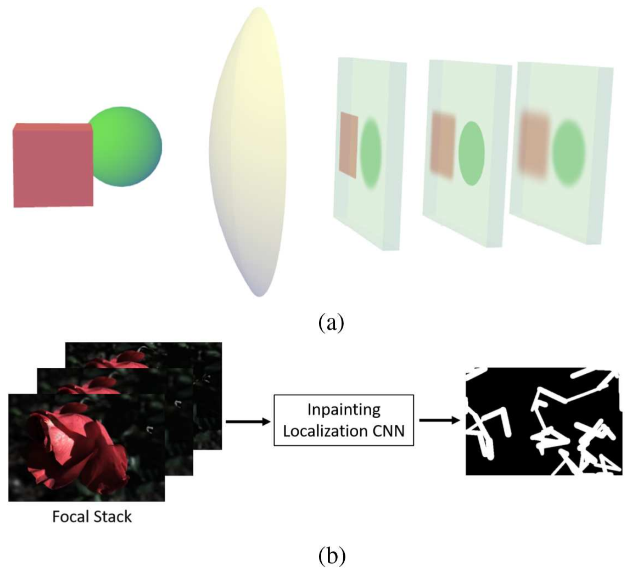

Fig. 1. Focal stack system for inpainting region localization. (a) Imaging system schematic showing depth-dependent defocus blur of a cube-ball object. (b) Inpainting localization CNN estimates inpainting regions from a focal stack.

focal stack is extremely challenging, we show that detecting image manipulation is much easier using a focal stack compared to using a single image, by using such inter-focal stack consistency cues. This approach leads to a much more secure media format. Someone attempting to manipulate the image would have to modify every image in the focal stack, and it would be extremely challenging to accomplish this in a way where the consistencies of the content and the defocus blur are maintained across the focal stack. Note that the proposed method is not for forgery detection of single 2D images, which is probably too easy to fake. The future of secure imaging could possibly rely on novel image representations rather than on single 2D images. We show that using the focal stack as a novel secure image format, to the best of our knowledge, substantially improves the performance of forgery detection compared to using a single conventional 2D image.

To demonstrate the advantage of focal stack image sets over single 2D images as a tamper-evident image file, we limit our scope to inpainting types of image manipulation. We generated inpainted focal stacks using several convolutional neural network (CNN) based methods [16–18]; we then trained inpainting region localization CNNs to detect regions in the focal stack that are inpainted. We show that the focal stack based method achieves significantly better detection performance and generalization ability, compared to single image based methods. We further study how detection performance depends on the number of images in the focal stack and also whether the performance gain of using a focal stack might be mainly due to increased total pixel number.

The paper is organized as follows. Section 2 describes related work on image inpainting, forgery localization, and focal stack cameras. Section 3 describes the method we used to generate inpainted focal stacks and the method to localize inpainted regions. Section 4 presents multiple numerical experiments and results. Finally, Section 5 gives a summary and concluding remarks.

2. RELATED WORK

A. Image Inpainting

Traditional image inpainting methods work well on highly textured or patterned regions, but fail on inpainted regions with rich context and semantic meaning, such as natural scenes and human faces. Simakov *et al*. proposed a bidirectional similarity measure, a metric based on nearest neighbor patch search, to determine if two signals are similar and can be used as the objective function for image inpainting. PatchMatch [19] accelerated the patch matching process in the bidirectional similarity measure using random search and coherence propagation. Shift-Map [20] achieved inpainting by computing a shift-map, where the pixels in the inpainting region are sampled from a relative position indicated by the shift-map. The shift-map is estimated by a global optimization objective function that contains a data term and a smoothness term. The optimization is done in a hierarchical way to accelerate the computation, with a low-resolution shift-map estimated first and then refined by a high-resolution one.

Deep learning based inpainting methods have better performance for inpainting complex objects and scenes due to their powerful capability for modeling the high level semantics presented in the image. The context encoder [21] is an early approach to image inpainting using deep learning methods. An encoder extracts semantic information from a masked input image, and a decoder reconstructs a full image with coherent contents filled in the inpainting region. Pixel-wise reconstruction loss and adversarial loss are used as the loss function to train the network. Later works typically follow this adversarial training to improve the fidelity of the inpainted region. Generative multi-column convolutional neural networks (GMCNNs) [17] uses a multi-column network to inpaint missing regions at multiple-scale in parallel. A confidence driven pixel reconstruction loss is used to constrain filling boundary pixels more strictly, compared to those pixels that are far away from the boundary. A Markov random fields (MRF) type regularization promotes content diversity in the inpainting region. As a standard convolution's response is conditioned on both valid pixels and placeholder values in the inpainting region, it also leads to color discrepancies. To resolve this issue, Liu *et al*. [22] proposed partial convolution to reduce these artifacts by introducing a layer-wise binary valid mask to select out only valid pixels for convolution computation and to normalize the convolution output. Gated Convolution [18] further generalized the partial convolution by having a learnable gating mechanism to select only proper pixels for convolution. Nazeri *et al*. [16] divided the inpainting process into edge generation and colorization stages. In the first stage, the edges of the inpainting regions are first generated. Then the colorization network inpaints the region conditioned on the input image and also the edge map. Such proposed two-stage inpainting exhibits better details in the inpainting region. There has been continued progress on improving inpainting using deep learning methods. Li *et al*. proposed to use a recurrent feature reasoning module to improve the inpainting performance on large continuous holes. Yi *et al*.

proposed a contextual residual aggregation mechanism to inpaint ultrahigh-resolution images with good quality [23]. Peng *et al*. proposed to use a hierarchical vector quantized variational auto-encoder (VQ-VAE), to generate diverse inpainting results [24].

B. Forgery Localization

Early attempts to localize manipulated regions in images relied on local anomalies of some signatures present in the image. Johnson *et al*. [9] analyzed the chromatic aberration presented in the image and identified image regions where chromatic aberrations are inconsistent with other regions in the image. Popescu *et al*. [7] showed that the color interpolation algorithm used for the color filter array in commercial cameras leads to periodic correlation patterns that can be revealed by Fourier analysis. They demonstrated that this signature can be used to localize tampered regions in an image. Assuming a known camera model or other reference images available, sensor pattern noise can also be used to localize a forged region by checking whether a region has such noise patterns [8]. In addition, splicing and copy-move forgery likely involves several post-processing steps, such as scaling/rotating the object and blurring the object/background boundary. These steps can generate re-sampling artifacts and can also be detected by spectral analysis [25].

Recent deep learning based methods, in contrast, learn discriminating forgery features from the data directly. Salloum *et al*. [26] trained a multi-task CNN (MFCN) for splicing localization. The network estimates both the splicing region and the splicing boundaries, with partially shared parameters between two tasks. Such multi-task design leads to better localization performance, compared to estimating only the splicing region. Huh *et al*. detected image splicing by training a classifier to determine whether two image patches have exchangeable image file (EXIF) meta consistency [10]. Wang *et al*. [11] detected image warping manipulation by training a CNN on script-generated warped images in Photoshop. Wu *et al*. [12] proposed a two-branch CNN model (BusterNet) to localize copy-move forgery regions. Li *et al*. [13] localized inpainted regions by using a CNN model with the first few layers initialized as high-pass filters to enhance the inpainting traces. Despite these efforts, developing a well performing forgery detection method with good generalization ability remains as a challenge.

C. Focal Stack

Recently, a focal stack camera employing transparent sensor arrays has been introduced that enables focal stack capture in a single camera exposure [27,28]. For static or sufficiently slow-moving scenes, focal stacks may also be captured by sequential exposure with refocusing by a conventional camera, or be synthesized from a light field using the add-shift algorithm [29]. There are numerous applications of focal stack imaging. Lien *et al*. [27] demonstrated model based light field reconstruction from focal stacks and 1D ranging. Zhang *et al*. [28] demonstrated 3D object localization and orientation estimation from a focal stack. Hazirbas *et al*. [30] trained a CNN to estimate depth maps from focal stack images. To the best of our knowledge, there is no prior work using focal stacks for image forensic related applications, and this work is the first to propose using focal stack imaging as a secure image format.

3. METHOD

To demonstrate the effectiveness of using focal stacks as a secure image format, we generated datasets containing manipulated focal stacks and trained a detection CNN to localize the forgery regions. The localization performance is then compared with single image based methods to show the advantage of focal stacks over conventional images for image security applications. We focus on image inpainting forgery where the inpainting is done by deep learning methods. Section 3.A describes how we generate inpainted focal stacks using CNN methods. Section 3.B describes how we localize inpainting regions in the manipulated focal stack.

A. Generating CNN Inpainted Focal Stack

We first generated a set of authentic focal stacks from the Lytro flower light field dataset [31], using the add-shift algorithm [29]. The Lytro flower light field dataset contains 3343 light fields of flower scenes captured by the Lytro Illum light field camera. Each light field has a size of 376 × 541 × 14 × 14, and following [31], we used only the central 8 × 8 sub-aperture images for focal stack generation. Each generated focal stack contains $n_{\mathrm{F}} = 7$ images with differing focus positions. The focus positions are chosen to have their corresponding disparities evenly distributed in range $[-1, 0.3]$, which covers roughly the entire possible object depth range. The first row of Fig. 2 shows example generated authentic focal stacks images.

Then we generated inpainted focal stack datasets, using three CNN based methods: GMCNN [17], EdgeConnect [16] and Gated Convolution [18]. GMCNN uses a multi-column network to extract features at different scale levels. A special implicit diversified Markov random field (ID-MRF) loss is designed to promote the diversity and realism of the inpainted region. EdgeConnect is a two-stage inpainting process. In the first stage, an edge generator generates edges for the inpainting region. In the second stage, an inpainting network fills the missing region with the help of the completed edges from the first stage. Gated Convolution [18] uses a learnable feature gating mechanism to solve the issue in which a normal convolution treats all pixels equally and inpaints the image following a two-stage coarse to fine process. We generate inpainted focal stacks using multiple methods to test the generalization ability of the network; we train the detection network using focal stacks inpainted by one method and then evaluate its performance on focal stacks inpainted by another method. This investigation mimics the more realistic scenario where the method used to inpaint the focal stack is unknown at the time of detection.

We generated random stroke-type regions to be inpainted for each focal stack. All images in the same focal stack shared the same spatial inpainting region. The goal of inpainting is typically trying to hide something in the original image and hence identical inpainting regions across images in the same focal stack should be a reasonable assumption. Each image is then inpainted independently using one of the above CNN methods.

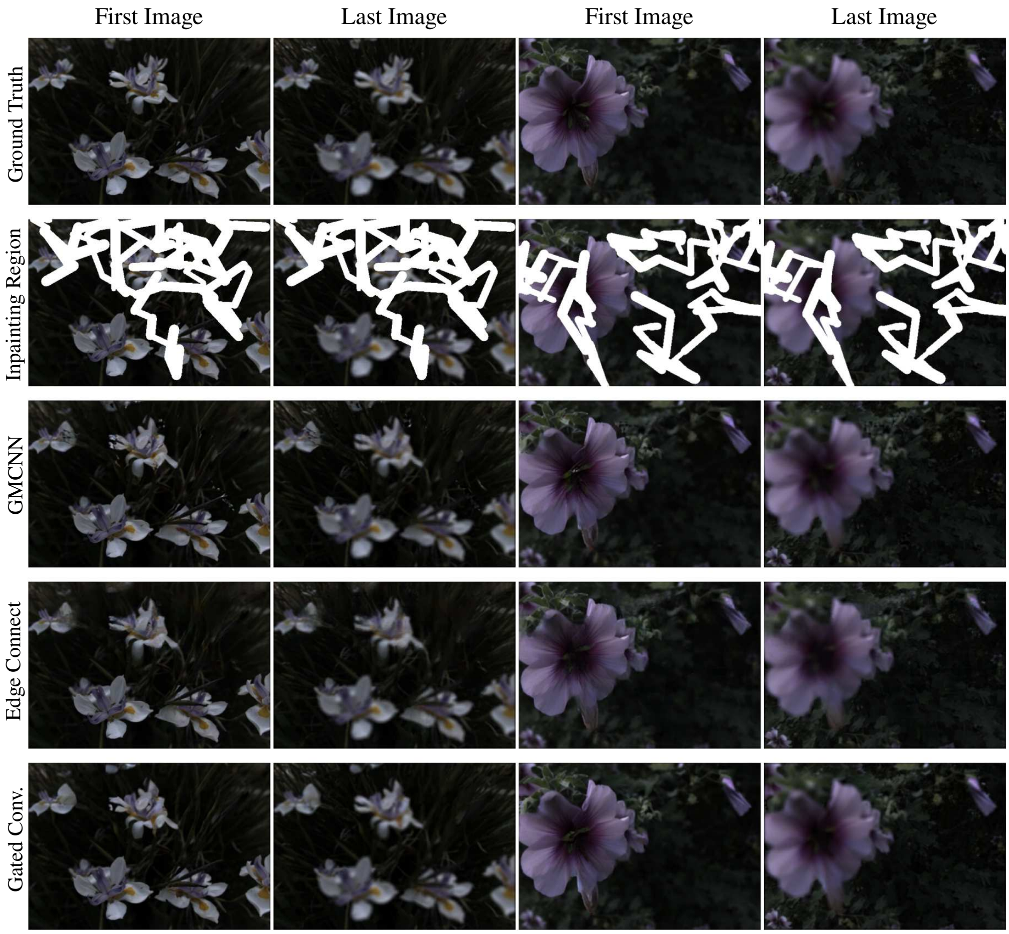

Fig. 2. Example real and inpainted focal stacks. Only the first and last images in each focal stack are shown. The region to be inpainted is shown as white in the second row.

The CNN inpainting models were pre-trained on the places2 [32] dataset using its original implementation and fined tuned on the flower focal stack dataset. Figure 2 shows example inpainted focal stacks.

B. Detecting CNN Inpainted Focal Stack

The detection network we used for localizing inpainting regions is based on DeepLabv3 [33]. DeepLabv3 was originally proposed for semantic segmentation, and we re-purposed it for region localization due to the similarity in these two tasks. The Atrous Spatial Pyramid Pooling (ASPP) layer in DeepLabv3 ensures a large receptive field and fine detailed network output at the same time, which is beneficial for our inpainting region localization. We used ResNet-18 [34] as the backbone for feature extraction. A normal input image to the DeepLabv3 is a 3D tensor of shape (C, H, W), whereas the focal stack is a 4D tensor of shape (n_F, C, H, W), so we reshaped the focal stack to be $(n_\mathrm{F} \times C, H, W)$ by concatenating images along the color channel. The network outputs a pixel-wise probability map that indicates whether a pixel is inpainted, and we train the network using binary cross-entropy loss.

Wang *et al.* [35] showed that proper data augmentations, such as applying JPEG compression, lead to a model with better generalization ability and robustness against common post-processing. Motivated by this, we followed their approach and trained our detection network with JPEG augmentation. Specifically, the training input focal stacks have a 50% probability of being JPEG compressed, with a JPEG quality factor of 70. For reference, we also trained models without JPEG augmentation; these models performed worse, and we include these results at the end of the paper.

4. EXPERIMENTS AND RESULTS

A. Implementation

1. Dataset

The inpainted focal stack dataset generated from Lytro flower light fields contains 3343 focal stacks for each inpainting method (GMCNN, EdgeConnect, Gated Convolution). Each focal stack contains $n_\mathrm{F} = 7$ images with changing focus depths and is associated with a ground truth inpainting region for training and evaluation. We used 2843 focal stacks for fine-tuning the inpainting networks and also training the detection network. The remaining 500 focal stacks are used for evaluating the inpainting localization performance.

2. *Training set-up*

We trained the detection network using the Adam optimizer [36] with batch size three. The models were trained for 110 epochs, with an initial learning rate 10^{-4} that was reduced to 10^{-5} after 70 epochs. We used data augmentation in the form of horizontal flipping with 50% probability, in addition to the JPEG compression augmentation described above.

3. *Evaluation*

We counted the true positive (TP), false positive (FP), and false negative (FN) predictions at the pixel level for each test sample, with the classification probability threshold set to 0.5. Then the F_1 scores, defined as $\frac{\mathrm{TP}}{\mathrm{TP}+\frac{1}{2}(\mathrm{FP}+\mathrm{FN})}$, were computed and averaged over all test samples to evaluate the network's inpainting localization performance.

We additionally tested the models' robustness against common post-processing methods including JPEG compression, gaussian noise, and resizing. Specifically, we added additive white gaussian noise with σ in range [0, 1.6] to test the robustness against noise. We downsampled test focal stacks using nearest neighbor interpolation with ratio in range [1,2] to test the robustness against resizing. We JPEG compressed test focal stacks with JPEG quality in range [30,100] to test the robustness against compression. Note that these post-processing processes are applied only to the test focal stacks; the models were trained using augmentation based only on horizontal flipping and JPEG compression with quality 70.

To study the dependence of localization performance on focal stack size n_F, we trained models using inpainted focal stack datasets with $n_\mathrm{F} = 1, 2, 3, 5, 7$. Specifically, the $n_\mathrm{F} = 7$ dataset is the one described in Section 4.A.1. We obtained the $n_\mathrm{F} = 1$ dataset by only using the 7th (last) image of each focal stack in $n_\mathrm{F} = 7$ dataset. Similarly, the $n_\mathrm{F} = 2$ dataset contains the first and seventh images, the $n_\mathrm{F} = 3$ dataset contains the first, fourth, and seventh images, and the $n_\mathrm{F} = 5$ dataset contains the first, third, fourth, fifth, and seventh images.

B. Results

Figure 3 shows the localization results trained on the GMCNN inpainted focal stack dataset and evaluated on testing focal stacks inpainted by GMCNN, EdgeConnect, and Gated Convolution. The advantage of using a focal stack ($n_\mathrm{F} \geq 2$) over a single image ($n_\mathrm{F} = 1$) for inpainting region localization is apparent and significant for every test configuration. Taking the first row of Fig. 3 for example, both training and testing on the GMCNN dataset using $n_\mathrm{F} = 1$ have a F_1 score of about 0.67 and using $n_\mathrm{F} = 2$ have a F_1 score of about 0.87. The difference is even more dramatic when training is performed on the GMCNN dataset and testing is performed on the Gated Convolution dataset (top-right subplot): $n_\mathrm{F} = 1$ has a F_1 score of about 0.11 and using $n_\mathrm{F} = 2$ has a F_1 score of about 0.80. Increasing n_F further improves the F_1 score, though not significantly. Although the single image ($n_\mathrm{F} = 1$) localization method performs fairly well when the testing data are generated by the same inpainting method as the training data, it performs poorly when the testing data are inpainted by a different method. On the other hand, *there is only a very small performance drop for the focal stack based method when testing on focal stacks inpainted by a method different from training*. These results show that the focal stack based method has a much better generalization ability across different inpainting methods. This benefit can be understood as follows: for single image based inpainting region localization, the network relies heavily on detecting inpainting method specific artifacts, such as checkerboard patterns produced by transpose convolutions [37] or unnatural transitions between inpainted and not inpainted regions, to determine whether a region is inpainted. However, these criteria cannot be universal for detecting inpainting because a different method will likely have a different checkerboard pattern or a different

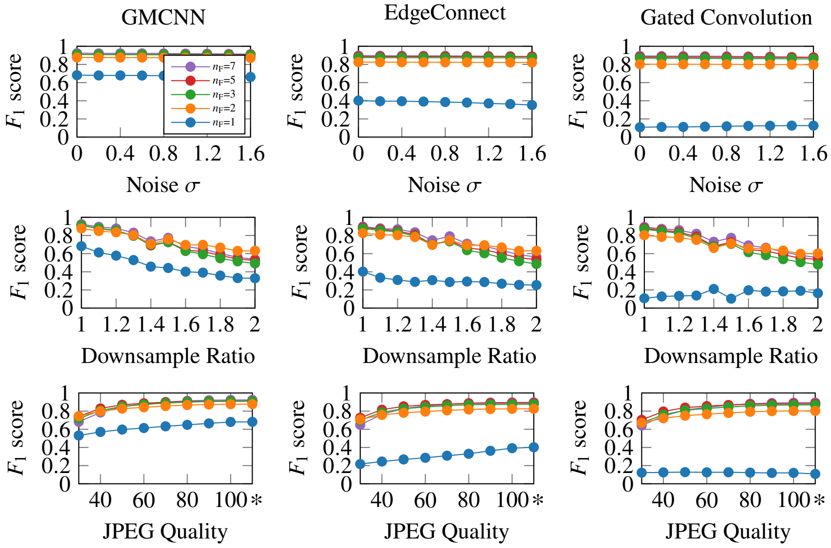

Fig. 3. Localization F_1 scores for focal stack data with the network trained on (Lytro flower) GMCNN dataset with JPEG augmentation and tested on (Lytro flower) GMCNN data (first column), EdgeConnect (second column), and Gated Convolution (third column) datasets. The robustness against Gaussian noise (first row), resizing (second row), and JPEG compression (third row) are shown for each model. Symbol * on x axis indicates the result without JPEG compression.

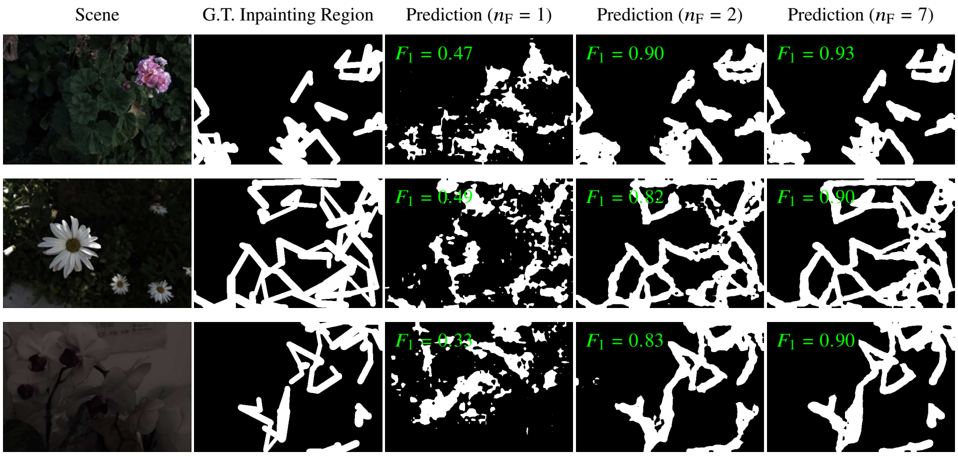

Fig. 4. Example localization results of the model trained on GMCNN dataset and tested on Gated Convolution dataset. Probability threshold of 0.5 is used for classification. F_1 scores are indicated in green for each prediction.

transition artifact between inpainted and not inpainted regions. On the other hand, the focal stack based method has a much more inpainting method agnostic clue to determine whether a region is inpainted or not: it can check whether the content and the defocus blur across a focal stack in a region are physically and semantically consistent. Such consistency checks do not depend on the methods used for inpainting, and hence it should better generalize across different inpainting methods.

Figure 4 shows example predicted inpainting regions, using a model trained on GMCNN inpainted focal stacks and tested on Gated Convolution inpainted focal stacks. The single image based inpainting localization performs poorly, whereas using a focal stack of only $n_F = 2$ greatly improves the prediction, and the $n_F = 7$ model has the best performance.

We also trained models using EdgeConnect inpainted focal stacks, and using Gated Convolution inpainted focal stacks, to verify that the trends above are not specific to the particular training dataset. Figures 5 and 6 show the results. The general findings are similar to those in Fig. 3, with some minor differences: the advantage of a focal stack over a single image for the model trained and tested on the EdgeConnect inpainted dataset is smaller, as shown in the middle column of Fig. 5. This is likely because the EdgeConnect inpainted images contain more visually apparent inpainting artifacts. Indeed, when we inspect closely some EdgeConnect inpainted regions, they tend to be darker, compared to non-inpainted regions. This makes inpainting localization using a single image easier, so the additional images in the focal stack do not help much. However, when the model is evaluated on the dataset inpainted by a method different from the training data, the single image localization performance degrades severely, as shown in the first and third columns of Fig. 5, while the focal stack based models retain high performance in these cases. This is again because the focal stack based method uses the more generalizable inter-focal stack consistency check to localize the inpainting region. For

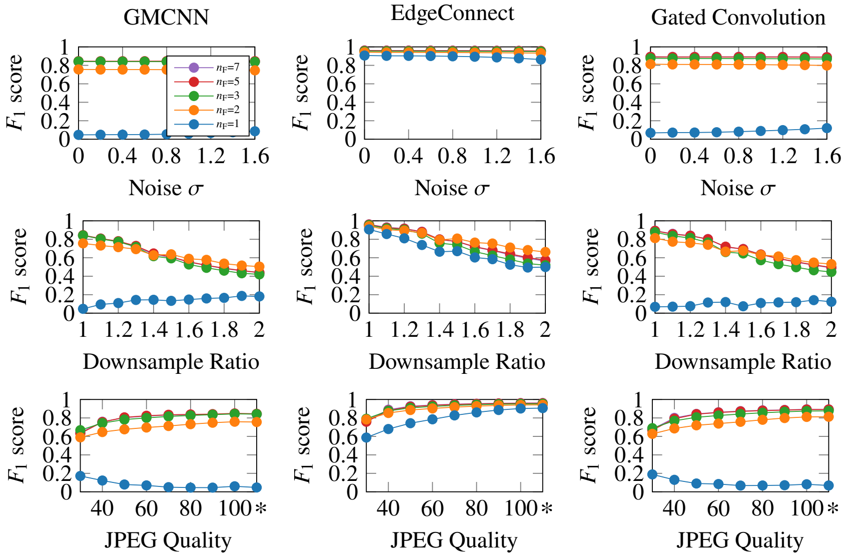

Fig. 5. Localization F_1 scores for focal stack data with the network trained on (Lytro flower) EdgeConnect dataset with JPEG augmentation and tested on (Lytro flower) GMCNN (first column), EdgeConnect (second column), and Gated Convolution (third column) datasets. Symbol * on x axis indicates the result without JPEG compression.

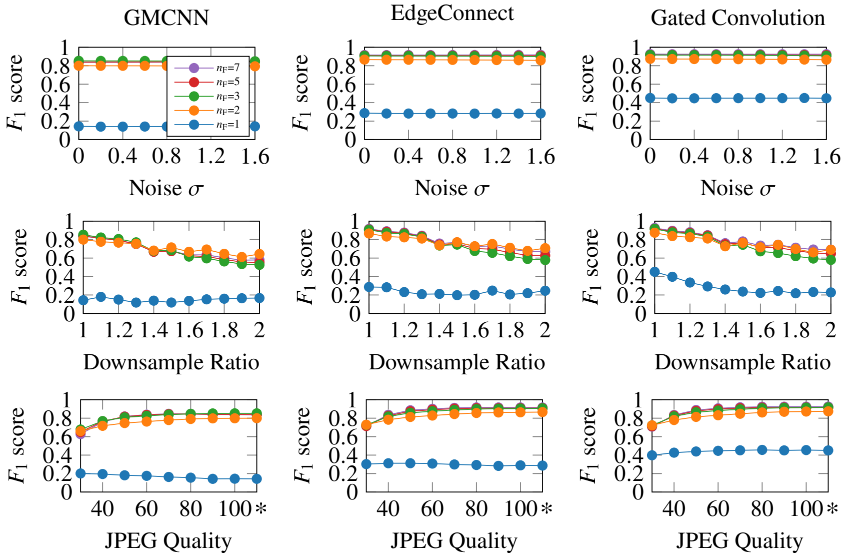

Fig. 6. Localization F_1 scores for focal stack data with the network trained on (Lytro flower) Gated Convolution dataset with JPEG augmentation and tested on (Lytro flower) GMCNN (first column), EdgeConnect (second column), and Gated Convolution (third column) datasets. Symbol * on x axis indicates the result without JPEG compression.

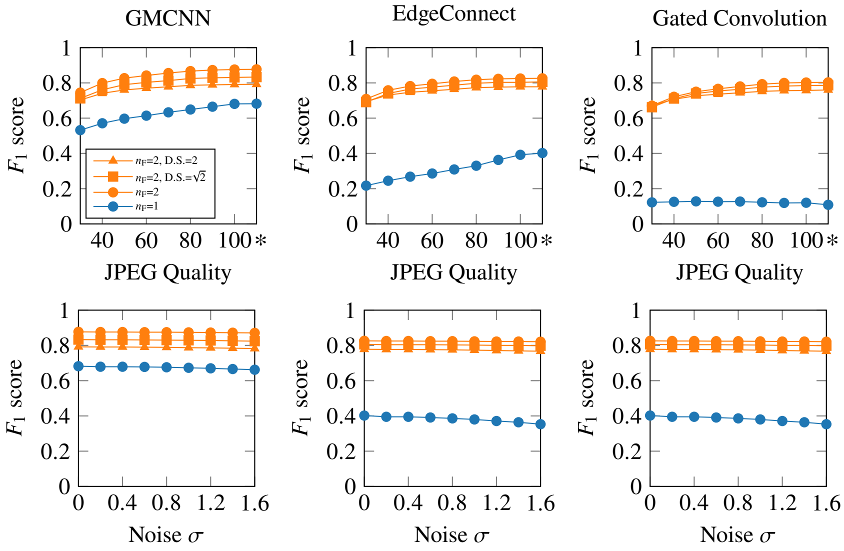

Fig. 7. Localization F_1 scores for focal stack data with the network trained on (Lytro flower) GMCNN dataset with JPEG augmentation and tested on (Lytro flower) GMCNN (first column), EdgeConnect (second column), and Gated Convolution (third column) datasets, showing the total pixel dependence. Symbol * on x axis indicates the result without JPEG compression.

models trained on Gated Convolution, the single image based method performs poorly (third column of Fig. 6), even when tested on focal stacks inpainted by the same method. This is because the Gated Convolution inpainted images contain fewer artifacts and are more visually realistic. This makes the single image based method struggle to find discriminating forgery traces.

All results presented in Fig. 3, Fig. 6, and Fig. 5 demonstrate good robustness against several post-processing methods, including Gaussian noise (first row), image resizing (second row), and JPEG compression (third row), showing that our proposed method would be useful in practical cases, such as in determining whether an Internet image file is authentic or not, where these post-processing operations are common.

To verify that the advantage of a focal stack over a single image is not simply due to the increase in the number of total pixels, we trained additional models for $n_F = 2$, using focal stacks downsampled by factors of $\sqrt{2}$ and two. Figure 7 shows the results. The $n_F = 2$, downsampling ratio $= \sqrt{2}$ system has the same total number of pixels as the $n_F = 1$ system without downsampling, and the $n_F = 2$, downsampling ratio $= 2$ model has two times fewer total pixels, compared to the system of $n_F = 1$, without downsampling. Figure 7 shows that reducing the total pixel numbers in the focal stack system only slightly reduces the localization performance; the main performance gain of using a focal stack for inpainting localization is due to the multiple sensor plane nature of the focal stack system that encodes robust inter-focal stack consistency clues for forgery detection.

Table 1. F_1 Scores of the Model Trained on GMCNN Inpainted Focal Stacks with Focusing Disparity Range [−1, 0.3], and Evaluated on Focal Stacks Inpainted by GMCNN, EdgeConnect, and Gated Convolution[a]

n_F	GMCNN	EdgeConnect	Gated Convolution
1	0.68/0.66/0.66	0.40/0.37/0.37	0.11/0.10/0.10
2	0.88/0.87/0.87	0.83/0.82/0.81	0.80/0.79/0.79
3	0.91/0.91/0.85	0.88/0.87/0.82	0.87/0.86/0.80
5	0.91/0.92/0.89	0.89/0.89/0.86	0.88/0.89/0.85
7	0.92/0.92/0.90	0.90/0.89/0.87	0.89/0.89/0.87

[a] Three values in each field correspond to the results on focal stacks with focusing disparity ranges [−1, 0.3], [−0.8, 0.5], and [−1.2, 0.5], respectively.

In practical applications, the testing focal stack to be authenticated may have a focus setting different from the training time focus setting. Thus, in Table 1, we also evaluate our model using inpainted focal stacks having a different focus setting compared to the training time. Specifically, the model is trained using GMCNN inpainted Lytro flower focal stacks, with focusing disparity evenly distributed in range [−1, 0.3], and tested on Lytro flower focal stacks with focusing disparity evenly distributed in range [−1, 0.3] (same setting as training), and in ranges [−0.8, 0.5] and [−1.2, 0.5]. The case of [−0.8, 0.5] corresponds to the scenario where every image in the testing focal stack is focusing closer to the camera, and the case of [−1.2, 0.5] corresponds to the scenario where the focus depth range is larger for the testing data compared to the training data. The table shows that there is

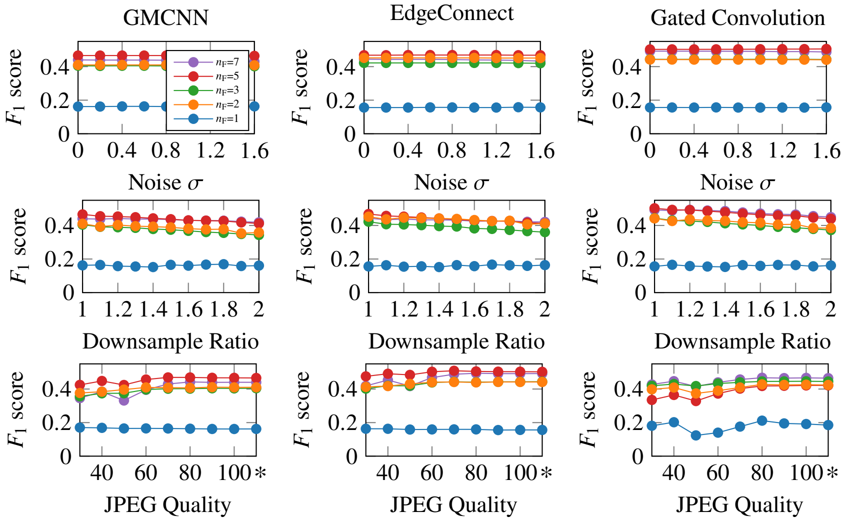

Fig. 8. Localization F_1 scores for focal stack data with the network trained on (Lytro flower) Gated Convolution dataset with JPEG augmentation and tested on (DUTLF) GMCNN data (first column), EdgeConnect (second column), and Gated Convolution (third column) datasets. The robustness against Gaussian noise (first row), resizing (second row), and JPEG compression (third row) are shown for each model. Symbol * on x axis indicates the result without JPEG compression.

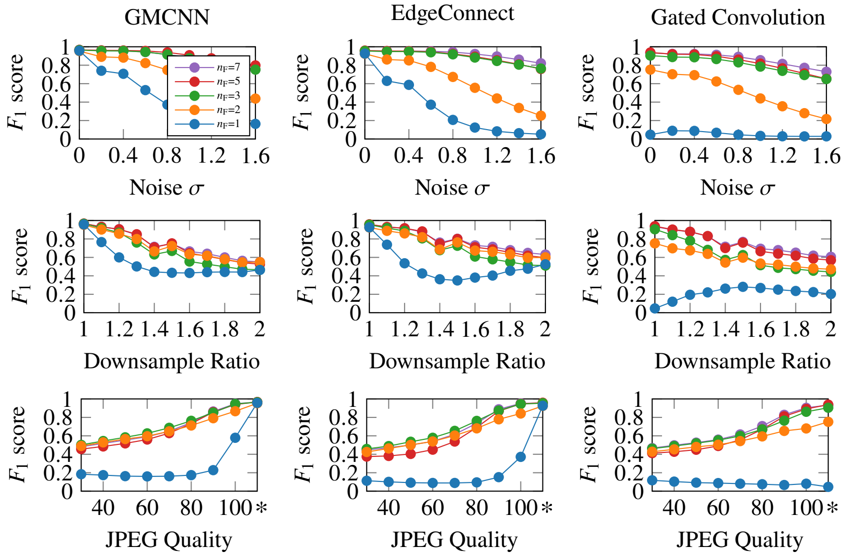

Fig. 9. Localization F_1 scores for focal stack data with the network trained on (Lytro flower) GMCNN dataset without JPEG augmentation and tested on (Lytro flower) GMCNN (first column), EdgeConnect (second column), and Gated Convolution (third column) datasets. Symbol * on x axis indicates the result without JPEG compression.

only a slight drop in inpainting localization performance when testing the trained focal stack based model on focal stacks with different focus settings. This excellent generalization ability across camera focus settings is due to the fact that the focal stack based model relies on the inter-focal stack consistency for detection, which is insensitive to the focus of each image.

To evaluate the model's generalization potential, i.e., its performance on an unseen dataset, we took the detection networks trained on the Lytro flower dataset with JPEG augmentation and tested their forgery localization performance directly on a new unseen dataset [38] [Dalian University of Technology Light Field (DUTLF)] without any additional network fine-tuning. The DUTLF dataset is a Lytro light field dataset, and the focal stack of each light field is also available. It is a challenging dataset with a diverse scene distribution. Figure S1 in Supplement 1 shows some example scenes. Figure 8 shows the generalization performance of the detection model trained on the Gated Convolution inpainted Lytro flower dataset, when evaluating on the inpainted DUTLF dataset. The F_1 scores of the focal stack based method are about twice as high as those of the single image based method ($n_F = 1$), demonstrating its superior generalization ability. Additional generalization performance experiments of the models trained on the GMCNN and on the EdgeConnect inpained Lytro flower dataset show similar results, and are included in Supplement 1 Figs. S2 and S3.

We also repeated the experiments in Figs. 3, 5, and 6 using the DUTLF dataset, and included these results in Fig. S4–6 of Supplement 1. We also found a dramatic localization accuracy gain when using the proposed focal stack based method, indicating that the performance improvement is not peculiar to a particular dataset.

Finally, to show the effect of JPEG augmentation during training, we include additional results of models trained without JPEG augmentation (Section 3.B). Comparing Figs. 3 and 9 shows that including JPEG augmentation during training leads to a model more robust against post-processing perturbations and better performance. The benefit is more significant for gaussian noise perturbation (first row of Fig. 9) and JPEG compression (third row of Fig. 9). The F_1 score of the model trained without JPEG augmentation will degrade quickly when the images are JPEG compressed or noise is added. Regardless, the advantage of using focal stacks over the single image based method is still significant for this training scheme as well.

5. CONCLUSION

We proposed a novel system and method, to the best of our knowledge, of using a focal stack for localizing image inpainting regions in manipulated images. We trained CNN models for inpainting localization and showed that using an image focal stack, instead of a single image, leads to significantly better localization performance and significant robustness to common post-processing image perturbations. The proposed method also shows excellent generalization ability across different inpainting methods and different camera focus settings.

Although we focused on the inpainting type of forgery, we expect the findings are applicable to many other types of forgery detection as well. We hope this work can lead to a new direction for image forgery detection and make images in the future more secure.

Funding. Division of Information and Intelligent Systems (1838179); W. M. Keck Foundation.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

Supplemental document. See Supplement 1 for supporting content.

REFERENCES

1. Deepfakes faceswap, https://github.com/deepfakes/faceswap.
2. P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recognition? Assessment and detection," arXiv:1812.08685 (2018).
3. H. Farid, "Image forgery detection," IEEE Signal Process. Mag. **26**(2), 16–25 (2009).
4. D. Cozzolino, D. Gragnaniello, and L. Verdoliva, "Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques," in *IEEE International Conference on Image Processing (ICIP)* (IEEE, 2014), pp. 5302–5306.
5. S. Dadkhah, A. Abd Manaf, Y. Hori, A. E. Hassanien, and S. Sadeghi, "An effective SVD-based image tampering detection and self-recovery using active watermarking," Signal Process. Image Commun. **29**, 1197–1210 (2014).
6. D. Singh and S. K. Singh, "Effective self-embedding watermarking scheme for image tampered detection and localization with recovery capability," J. Vis. Commun. Image Represent. **38**, 775–789 (2016).
7. A. C. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," IEEE Trans. Signal Process. **53**, 3948–3959 (2005).
8. J. Lukáš, J. Fridrich, and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," Proc. SPIE **6072**, 60720Y (2006).
9. M. K. Johnson and H. Farid, "Exposing digital forgeries through chromatic aberration," in *Proceedings of the 8th Workshop on Multimedia and Security* (2006), pp. 48–55.
10. M. Huh, A. Liu, A. Owens, and A. A. Efros, "Fighting fake news: image splice detection via learned self-consistency," in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 101–117.
11. S.-Y. Wang, O. Wang, A. Owens, R. Zhang, and A. A. Efros, "Detecting photoshopped faces by scripting photoshop," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2019), pp. 10072–10081.
12. Y. Wu, W. Abd-Almageed, and P. Natarajan, "BusterNet: detecting copy-move image forgery with source/target localization," in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 168–184.
13. H. Li and J. Huang, "Localization of deep inpainting using high-pass fully convolutional network," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2019), pp. 8301–8310.
14. X. Zhang, S. Karaman, and S.-F. Chang, "Detecting and simulating artifacts in GAN fake images," in *IEEE International Workshop on Information Forensics and Security (WIFS)* (IEEE, 2019), pp. 1–6.
15. D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva, "ForensicTransfer: weakly-supervised domain adaptation for forgery detection," arXiv:1812.02510 (2018).
16. K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, "EdgeConnect: structure guided image inpainting using edge prediction," in *IEEE International Conference on Computer Vision (ICCV) Workshops* (2019).
17. Y. Wang, X. Tao, X. Qi, X. Shen, and J. Jia, "Image inpainting via generative multi-column convolutional neural networks," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (2018), pp. 329–338.
18. J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *Proceedings of the*

IEEE International Conference on Computer Vision (ICCV) (2019), pp. 4471–4480.

19. C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, “PatchMatch: a randomized correspondence algorithm for structural image editing,” ACM Trans. Graph. **28**, 24 (2009).
20. Y. Pritch, E. Kav-Venaki, and S. Peleg, “Shift-map image editing,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2009), pp. 151–158.
21. D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 2536–2544.
22. G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, “Image inpainting for irregular holes using partial convolutions,” in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 85–100.
23. Z. Yi, Q. Tang, S. Azizi, D. Jang, and Z. Xu, “Contextual residual aggregation for ultra high-resolution image inpainting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 7508–7517.
24. J. Peng, D. Liu, S. Xu, and H. Li, “Generating diverse structure for image inpainting with hierarchical VQ-VAE,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 10775–10784.
25. A. C. Popescu and H. Farid, “Statistical tools for digital forensics,” in *International Workshop on Information Hiding* (Springer, 2004), pp. 128–147.
26. R. Salloum, Y. Ren, and C.-C. J. Kuo, “Image splicing localization using a multi-task fully convolutional network (MFCN),” J. Visual Commun. Image Represent. **51**, 201–209 (2018).
27. M.-B. Lien, C.-H. Liu, I. Y. Chun, S. Ravishankar, H. Nien, M. Zhou, J. A. Fessler, Z. Zhong, and T. B. Norris, “Ranging and light field imaging with transparent photodetectors,” Nat. Photonics **14**, 143–148 (2020).
28. D. Zhang, Z. Xu, Z. Huang, A. R. Gutierrez, C. J. Blocker, C.-H. Liu, M.-B. Lien, G. Cheng, Z. Liu, I. Y. Chun, J. A. Fessler, Z. Zhong, and T. B. Norris, “Neural network based 3d tracking with a graphene transparent focal stack imaging system,” Nat. Commun. **12**, 1–7 (2021).
29. R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, “Light field photography with a hand-held plenoptic camera,” Computer Science Technical Report CSTR 2005-02 (2005), pp. 1–11.
30. C. Hazirbas, S. G. Soyer, M. C. Staab, L. Leal-Taixé, and D. Cremers, “Deep depth from focus,” in *Asian Conference on Computer Vision* (Springer, 2018), pp. 525–541.
31. P. P. Srinivasan, T. Wang, A. Sreelal, R. Ramamoorthi, and R. Ng, “Learning to synthesize a 4D RGBD light field from a single image,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 2243–2251.
32. B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: a 10 million image database for scene recognition,” IEEE Trans. Pattern Anal. Mach. Intell. **40**, 1452–1464 (2017).
33. L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” arXiv:1706.05587 (2017).
34. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.
35. S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, “CNN-generated images are surprisingly easy to spot…for now,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8695–8704.
36. D. P. Kingma and J. L. Ba, “Adam: a method for stochastic optimization,” in *International Conference on Learning Representations (ICLR)* (2015), pp. 1–15.
37. A. Odena, V. Dumoulin, and C. Olah, “Deconvolution and checkerboard artifacts,” Distill **1**, e3 (2016).
38. T. Wang, Y. Piao, X. Li, L. Zhang, and H. Lu, “Deep learning for light field saliency detection,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2019), pp. 8838–8848.

Supplemental Document

applied optics

Focal stack based image forgery localization: supplement

ZHENGYU HUANG, JEFFREY A. FESSLER, AND THEODORE B. NORRIS*

Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, Michigan, 48105, USA
**Corresponding author: tnorris@umich.edu*

Supplement DOI: https://doi.org/10.6084/m9.figshare.19617372

Parent Article DOI: https://doi.org/10.1364/AO.450654

Supplement: Focal Stack Based Image Forgery Localization

ZHENGYU HUANG, JEFFREY A. FESSLER, AND THEODORE B. NORRIS[*]

Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI, 48105, USA

[*]*tnorris@umich.edu*

Supplementary figures referred to in the text are included herein.

The DUTLF dataset contains diverse scenes with a focal stack image resolution of 400×600. We used 400 DUTLF focal stacks for training and 100 for testing.

Fig. S1. Example scenes of the DUTLF dataset.

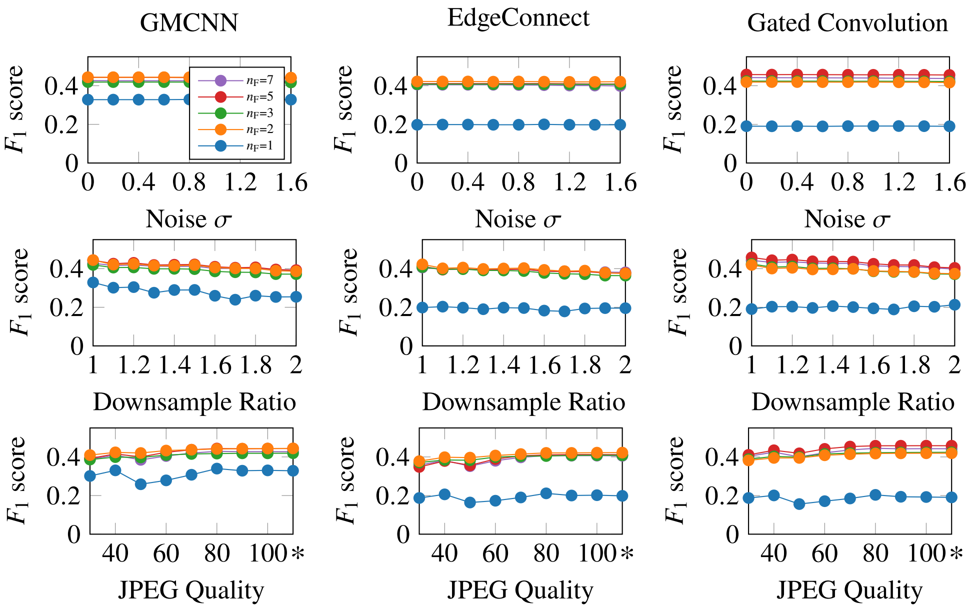

Fig. S2. Localization F_1 scores for focal stack data with the network trained on (Lytro flower) GMCNN dataset with JPEG augmentation and tested on (DUTLF) GMCNN data (1st column), EdgeConnect (2nd column) and Gated Convolution (3rd column) datasets. The robustness against Gaussian noise (1st row), resizing (2nd row) and JPEG compression(3rd row) are shown for each model. Symbol '*' on x-axis indicates the result without JPEG compression.

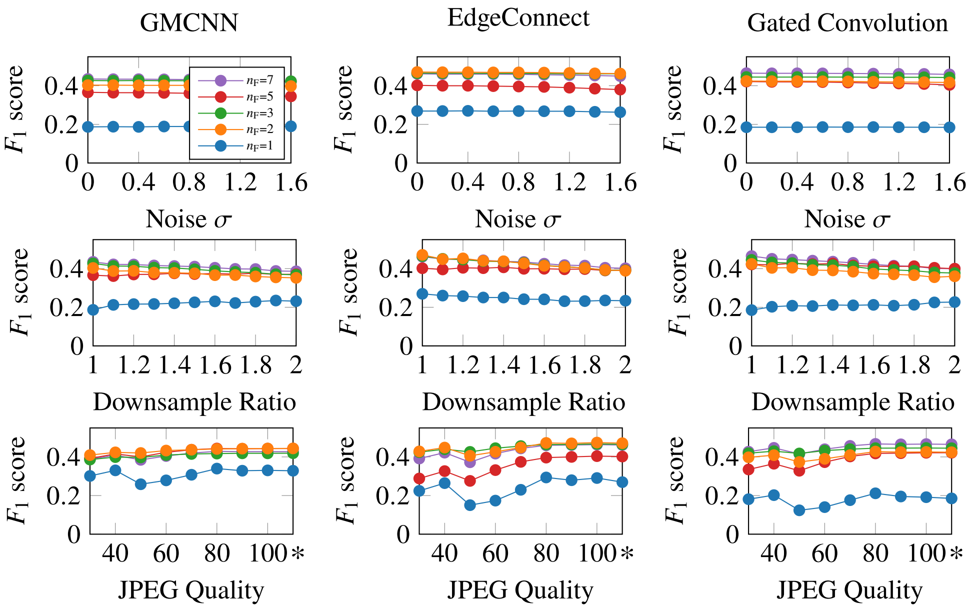

Fig. S3. Localization F_1 scores for focal stack data with the network trained on (Lytro flower) EdgeConnect dataset with JPEG augmentation and tested on (DUTLF) GMCNN data (1st column), EdgeConnect (2nd column) and Gated Convolution (3rd column) datasets. The robustness against Gaussian noise (1st row), resizing (2nd row) and JPEG compression(3rd row) are shown for each model. Symbol '*' on x-axis indicates the result without JPEG compression.

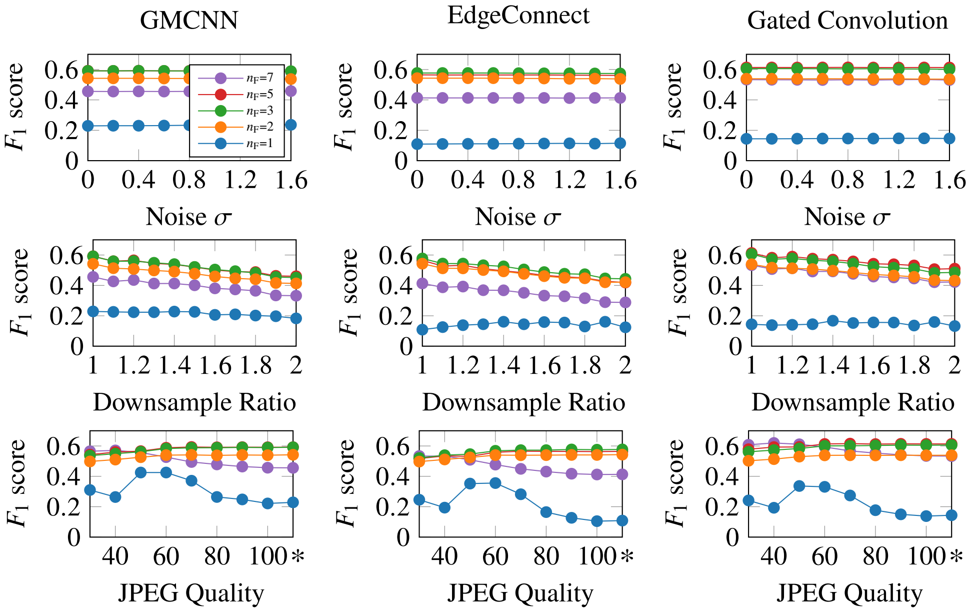

Fig. S4. Localization F_1 scores for focal stack data with the network trained on (DUTLF) GMCNN dataset with JPEG augmentation and tested on (DUTLF) GMCNN data (1st column), EdgeConnect (2nd column) and Gated Convolution (3rd column) datasets. The robustness against Gaussian noise (1st row), resizing (2nd row) and JPEG compression(3rd row) are shown for each model. Symbol '*' on x-axis indicates the result without JPEG compression.

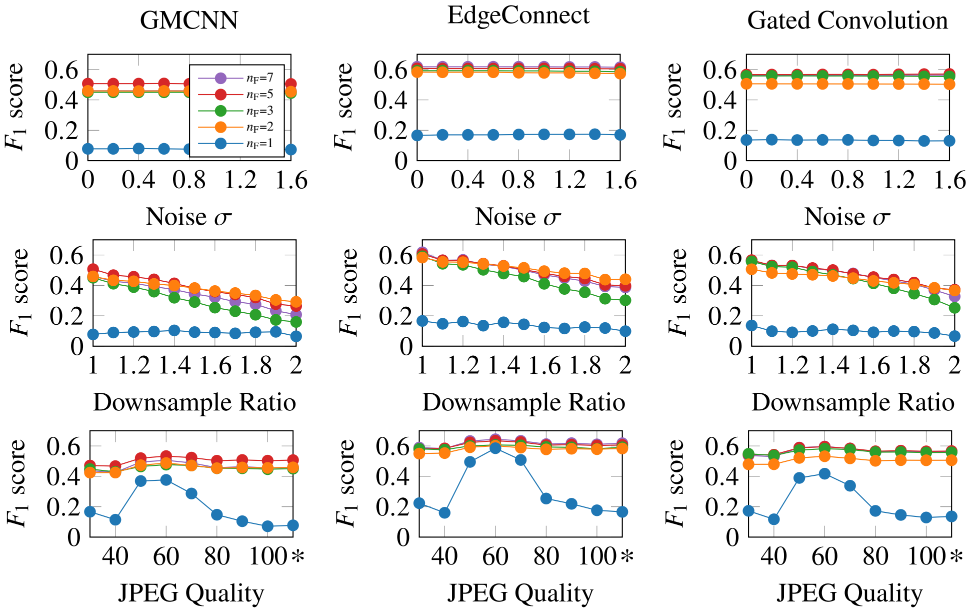

Fig. S5. Localization F_1 scores for focal stack data with the network trained on (DUTLF) EdgeConnect dataset with JPEG augmentation and tested on (DUTLF) GMCNN data (1st column), EdgeConnect (2nd column) and Gated Convolution (3rd column) datasets. The robustness against Gaussian noise (1st row), resizing (2nd row) and JPEG compression(3rd row) are shown for each model. Symbol '*' on x-axis indicates the result without JPEG compression.

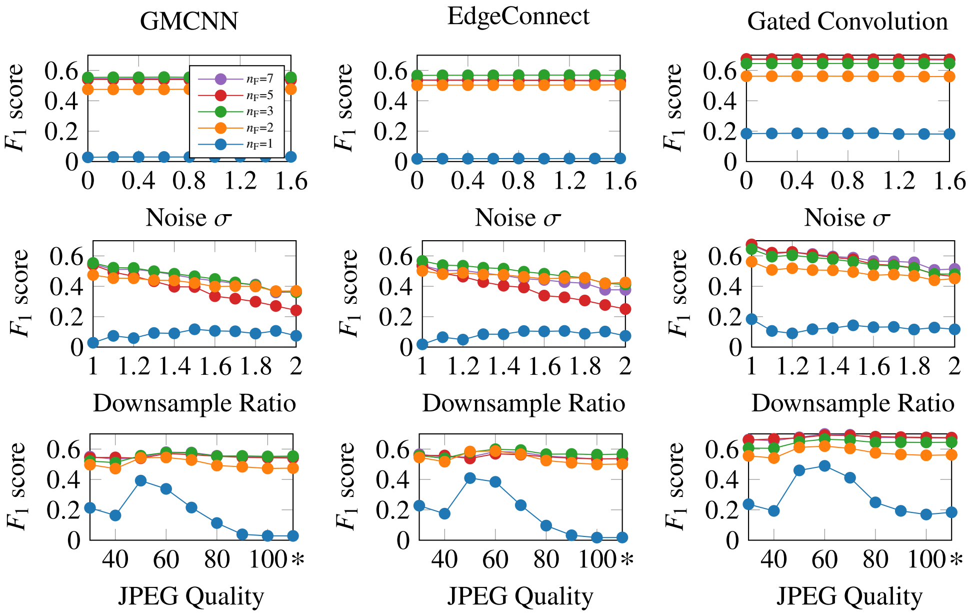

Fig. S6. Localization F_1 scores for focal stack data with the network trained on (DUTLF) Gated Convolution dataset with JPEG augmentation and tested on (DUTLF) GMCNN data (1st column), EdgeConnect (2nd column) and Gated Convolution (3rd column) datasets. The robustness against Gaussian noise (1st row), resizing (2nd row) and JPEG compression(3rd row) are shown for each model. Symbol '*' on x-axis indicates the result without JPEG compression.