CAT-Net: Compression Artifact Tracing Network for Detection and Localization of Image Splicing

Myung-Joon Kwon, In-Jae Yu, Seung-Hun Nam, and Heung-Kyu Lee
Korea Advanced Institute of Science and Technology (KAIST)
{kwon19,myhome9830,nam1202,heunglee}@kaist.ac.kr

Abstract

Detecting and localizing image splicing has become essential to fight against malicious forgery. A major challenge to localize spliced areas is to discriminate between authentic and tampered regions with intrinsic properties such as compression artifacts. We propose CAT-Net, an end-to-end fully convolutional neural network including RGB and DCT streams, to learn forensic features of compression artifacts on RGB and DCT domains jointly. Each stream considers multiple resolutions to deal with spliced object's various shapes and sizes. The DCT stream is pretrained on double JPEG detection to utilize JPEG artifacts. The proposed method outperforms state-of-the-art neural networks for localizing spliced regions in JPEG or non-JPEG images.

1. Introduction

Modern mobile devices mean that anyone can take a picture anywhere anytime. Image editing is very easy due to user-friendly image editing software and images can be shared in seconds due to social networking services. Although these advances have benefited people's lives, they have also caused problems when forged images are used as fake news, false propaganda, or fake evidence [44]. Therefore, it has become increasingly important to detect image manipulations.

Image splicing is defined as copy-pasting some part of an image onto another image (Figure 1) [37]. It is one of the easiest and most popular image manipulations, but it is also one of the most frequently used manipulations for bad purposes. For example, one can make a person appear to be somewhere they should not and had not been. Thus, this paper focuses on image splicing detection and localization. Given a possibly spliced image (Figure 1(c)), our goal is to generate a mask that localizes the potentially tampered image portion (Figure 1(d)).

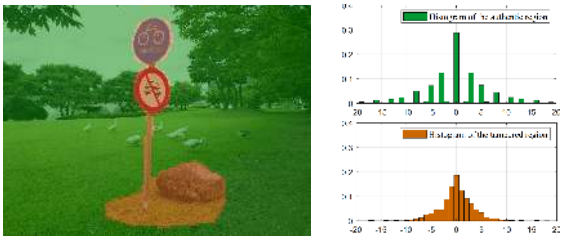
To distinguish between spliced and authentic areas, it is important to analyze statistical fingerprints caused by internal processes of camera or image editing software (*e.g.* sensor pattern noise [22], interpolation traces from the color filter array [30], compression artifacts [2, 26, 38, 40], *etc.*). Modern digital cameras typically compress the image to reduce storage space, with JPEG compression being employed in most cases due to its efficiency. However, this generates various JPEG artifacts due to information loss, even though they are generally not visible to human eyes. Thus, analyzing JPEG compression artifacts could help localize forged regions.

Double JPEG detection, *i.e.*, determining if a JPEG image has been compressed once or twice, can help identify splice forgery. A region spliced onto another image will likely have a statistically different distribution of DCT coefficients in Y-channel compared with an authentic region (Figure 2). The authentic region is doubly compressed: first in the camera and again as part of the forgery, leaving periodic patterns in the histogram [29]. The spliced

(a) Authentic image 1 (b) Authentic image 2 (c) Spliced image

(d) Ground truth mask (e) ManTra-Net [41] (f) CAT-Net (Proposed)

Figure 1. Challenge of localizing spliced regions from a JPEG image. Although ManTra-Net can trace various manipulations using RGB pixels, it is not ideal for capturing compression artifacts. The proposed approach considers RGB and DCT domains jointly to effectively tracks visual clues and compression traces.



(a) Spliced image (b) DCT histograms

Figure 2. Statistical differences between tampered and authentic regions. DCT histograms are obtained from Y-channel DCT coefficients at the frequency (2,1) for tampered and authentic regions separately.

region behaves like singly compressed, following the secondary quantization table [29]. Traditionally, DCT histograms have been employed to detect double JPEG compression [7, 21]. Even in deep learning era, deep neural networks tend to require preprocessed histograms as input [2, 26, 38, 40] because naively giving DCT coefficients as input commonly performs poorly due to the large decorrelation of DCT coefficients, unlike pixels [42]. All such methods produce patch-wise predictions due to the usage of histograms. Therefore, we adopt the binary volume representation for DCT coefficients to obtain pixel-wise predictions, originally designed for steganalysis [42]. This allows combining a semantic segmentation network with the double JPEG detection concept, providing pixel-wise prediction.

This paper proposes Compression Artifact Tracing Network (CAT-Net), an end-to-end fully convolutional neural network to detect and localize spliced regions. The network includes an RGB stream, a DCT stream, and a final fusion stage. The RGB stream learns visual artifacts and the DCT stream learns compression artifacts (*i.e.*, DCT coefficient distributions). We pretrain the DCT stream for double JPEG detection and use it as initialization for splicing localization. The fusion stage fuses multiple resolution features from the two streams to generate the final mask.

Our main contributions are summarized as follows.

- For the first time, CAT-Net localizes spliced objects considering RGB and DCT domains jointly. Extensive experiments with diverse benchmark datasets showed CAT-Net achieved state-of-the-art performance compared to baselines [41, 15], and stable performance for JPEG and non-JPEG images.
- We designed the DCT stream to learn compression artifacts that trace double-compressed clues based on binary volume representation of DCT coefficients. This approach outperforms previous state-of-the-art networks using histogram representation [2, 26, 40] in terms of detecting double JPEG compression.

2. Related Work

Image forgery localization can be categorized as block-wise classification, patch matching, and end-to-end neural network approaches.

Block-wise classification finds forgery distributions using classification per block for specific manipulations such as double JPEG compression [2, 26, 38, 40], image resampling [30], contrast enhancement [36], and multiple manipulations [4]. Images are divided into several fixed-sized blocks to localize manipulation areas, and detection results from each block are combined. Detection is performed independently for each block, hence overall image statistics cannot be derived.

Patch matching extracts statistical features from image patches and measures consistency among the patches. Highly inconsistent patches are considered to have been manipulated (*e.g.* spliced from another image). Predefined feature extractors [1, 34] or neural networks [15, 23] are used to extract appropriate features for matching. Huh *et al.* [15] proposed a self-supervised approach to train a model to determine whether an image was self-consistent in terms of EXIF metadata. However, patch matching localization requires high resources because it needs to compute consistency for every patch pair and it needs time-consuming post-processing to derive actual forgery location by aggregating results from all pairs.

Neural networks have improved object detection [18, 31, 32] and semantic segmentation [20, 33, 39] performance considerably, and hence image forgery localization methods have been developed employing such techniques. In [45], SRM kernel [11] was added to an object detection model to extract bounding boxes of splicing, copy-move, and removal forgeries. Bi *et al.* [5] proposed a U-Net [33] based segmentation network to localize image splicing. It only used RGB pixel domain information like usual semantic segmentation networks. Wu *et al.* [41] proposed ManTra-Net using SRM kernel [11] in feature extraction and constrained convolution [4] followed by pixel-wise anomaly detection. Although they considered JPEG compression as a type of manipulation to train the feature extractor, it was unable to distinguish single and double JPEG compression. Consequently, localization performance degrades for JPEG images.

We propose a novel approach to detect and localize image splicing on JPEG images, overcoming the limitations of previous works. For fast inference and obtaining pixel-wise prediction, we adopt a segmentation network considering multi-resolution features [39]. To make the network robust to JPEG compression, we extract JPEG artifacts in the DCT domain employing binary volume representation of quantized DCT coefficients [42].

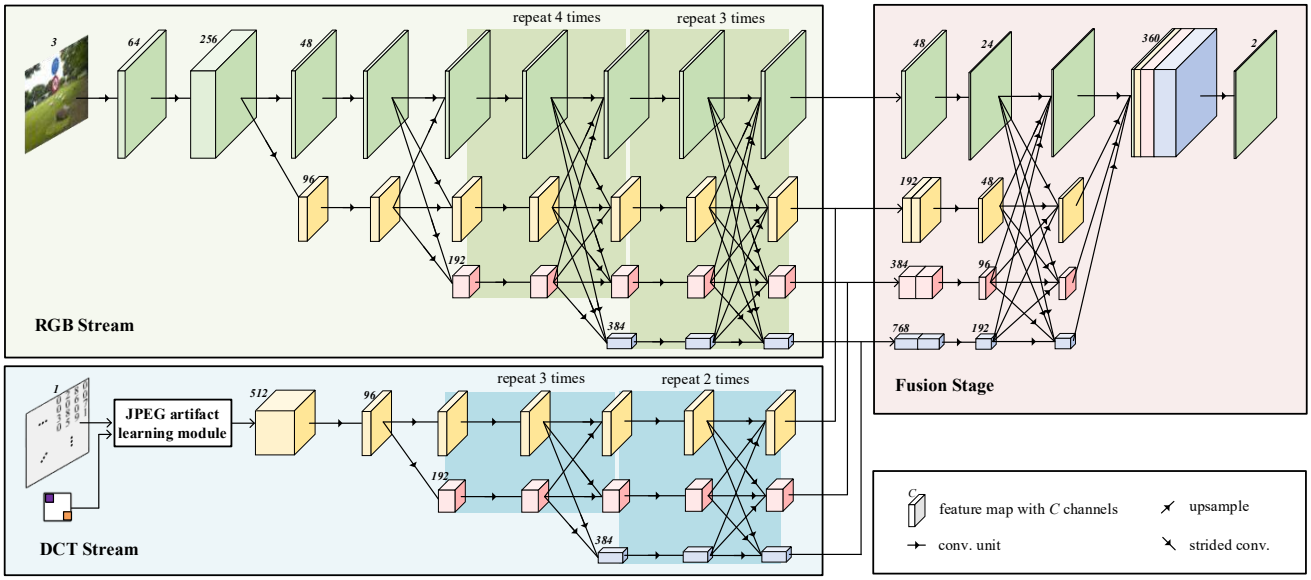
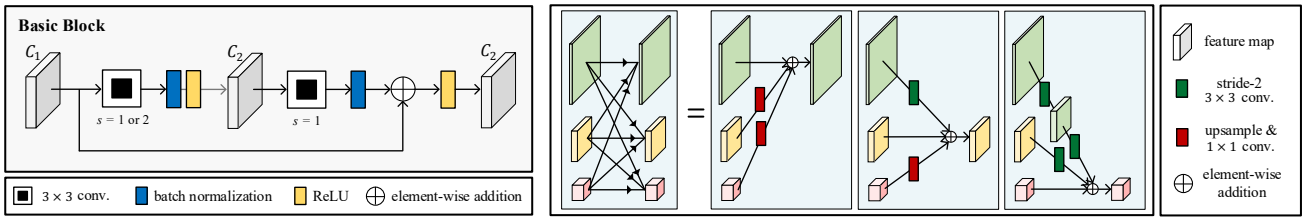


Figure 3. The proposed CAT-Net architecture includes an RGB stream, a DCT stream, and a final fusion stage. The RGB stream takes RGB pixels and the DCT stream takes Y-channel DCT coefficients and a Y-channel quantization table as input. The JPEG artifact learning module is shown in Figure 5.



(a) Basic block (b) Fusion unit

Figure 4. Elements in the proposed network. A convolutional unit in Figure 3 mainly consists of four consecutive basic blocks. The fusion unit fuses multi-resolution feature maps by summing them after matching resolutions.

3. Proposed Method

3.1. Network Structure

Figure 3 shows that CAT-Net comprises an RGB stream, a DCT stream, and a final fusion stage. RGB pixel values, quantized Y-channel DCT coefficients, and a Y-channel quantization table are extracted from JPEG file input. The RGB pixel values are fed into the RGB stream and the other data into the DCT stream. The RGB stream focuses on visual clues and the DCT stream on compression artifacts. The stream outputs are then fused to generate the final output.

We use HRNet [39] as the CAT-Net backbone, which was originally designed for computer vision problems. We introduce HRNet to a forensic problem since it maintains high-resolution representations through the whole process and employs a novel fusion method to combine multiple resolution features and capture the overall picture. This helps capture the overall structure without losing fine artifacts required for forensic investigations. Also, HRNet uses stride-2 convolution to downsample feature maps and does not use pooling layers. Recent studies have shown that pooling is undesirable for tasks that require subtle signals since pooling reinforces content and suppresses noise-like signals [6]. Although this behavior is desirable for computer vision tasks, it is inappropriate for forensic tasks since noise is an important clue.

The network includes two elements: a convolutional unit and a fusion unit. Each convolutional unit in Figure 3 consists of four consecutive basic blocks shown in Figure 4(a), with a few exceptions such as the first and the last part [39]. Figure 4(b) shows the fusion unit, which fuses multi-resolution feature maps by summing multi-resolution features after matching resolutions by bilinear interpolation (upsampling) or strided convolution (downsampling).

The RGB stream structure is identical to HRNet except the last part is removed. The RGB stream takes RGB pixel values as input and the first convolutional unit reduces res-

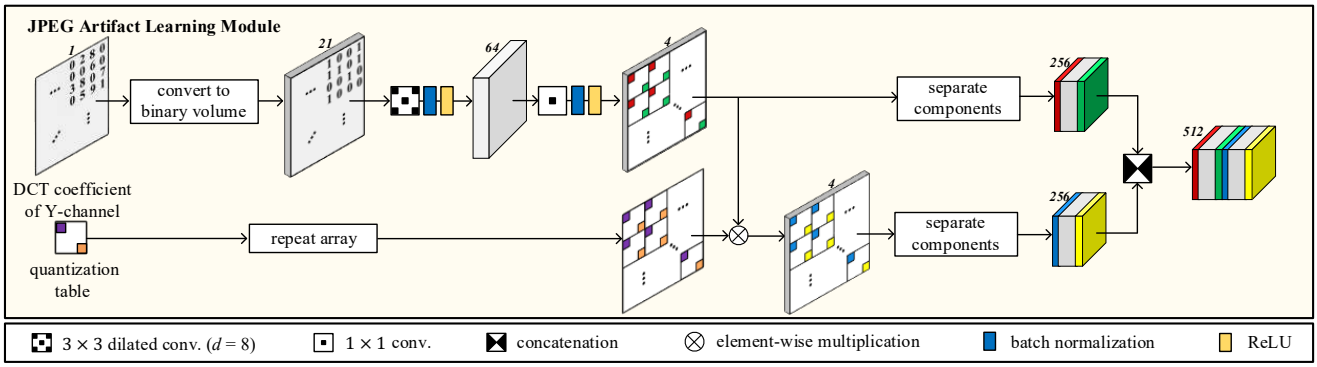


Figure 5. Proposed JPEG artifact learning module architecture.

olution 4-fold. Starting from the high-resolution path, it gradually goes through the network adding high-to-low resolution paths one by one and connecting multi-resolution paths in parallel. Each resolution remains until the end, producing $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, and $\frac{1}{32}$ resolutions.

The DCT stream captures compression artifacts, *i.e.*, statistical distributions of Y-channel DCT coefficients. The structure is a three-resolution variant of HRNet with the first convolutional unit replaced by a JPEG artifact learning module (Figure 5). All convolutional units in this stream comprise four basic blocks (Figure 4(a)) without exceptions.

The JPEG artifact learning module initially converts the input array of DCT coefficients, $\mathbf{M}$, to a binary volume using the transformation $f : \mathbb{Z}^{H\times W} \rightarrow \{0,1\}^{(T+1)\times H\times W}$ such that

$$f(\mathbf{M})_{t,i,j} = \begin{cases} 1, & \text{if } abs(clip(\mathbf{M}))_{i,j} = t \\ 0, & \text{otherwise} \end{cases}, \qquad (1)$$

where $clip(\cdot)$ clips the array element-wisely into the interval $[\ \ T, T]$ and $abs(\cdot)$ takes element-wise absolute values [42]. We experimentally determined optimal T to be 20. This binary volume representation is similar to DCT histogram [40] but allows the network to learn the relationship among adjacent DCT coefficients. DCT histogram merges information patch-wise, whereas this representation maintains image resolution which is suitable for segmentation.

Consecutive convolutions are applied to the binary volume. Dilated convolution is used here, which is originally designed for increasing CNN receptive fields [43]. However, the proposed network uses 8-dilated convolutions in order to extract features in DCT coefficients derived from the same frequency basis. The number of feature map channels is reduced to 4 using 1×1 convolution and the feature map is forked. For the forked path, 8×8 quantization table obtained from the JPEG header is multiplied to the corresponding frequency components. This is similar to the procedure of dequantizing DCT coefficients in JPEG decoding. For the other path, the table is not multiplied. Each 64 $(= 8 \times 8)$ frequency component is separated for both paths. Note that previous operations are done frequency-wise, hence each value in an 8×8 block represents a frequency component. Separating components changes shape from $4 \times H \times W$ to $256 \times \frac{H}{8} \times \frac{W}{8}$, which helps to significantly reduce resolution. Lastly in this module, the feature maps from the two paths are concatenated in channel dimension. The output passes the remaining path of the DCT stream.

During training, input images are cropped to a fixed size to construct a tensor having a batch dimension. It is worth noting that a rectangular cropping region must be aligned with the 8×8 grid since JPEG encodes images into 8×8 blocks. This makes each channel of a channel-separated tensor represent a frequency component. This also allows the RGB stream to learn JPEG blocking artifacts as well as visual artifacts.

Output feature maps have resolutions $(\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32})$ and $(\frac{1}{8}, \frac{1}{16}, \frac{1}{32})$ for the RGB and DCT streams, respectively. Two-stream feature maps are concatenated resolution-wise in channel dimension and passed to the final fusion stage (Figure 3), which is structurally identical to the final HRNet stage, but with a different number of channels. All four resolution feature maps are finally bilinearly upsampled to match the highest resolution, concatenated, and pass the final convolutional layer. The final output is a $2 \times \frac{H}{4} \times \frac{W}{4}$ array of logits for each class (authentic and tampered).

3.2. Handling non-JPEG images

Although our network uses a quantization table as input, the network can also handle non-JPEG images. Since non-JPEG images do not contain quantized DCT coefficients, they are calculated from RGB pixels, similarly to a JPEG encoder. We regard the quantization table for those images to be all ones, corresponding to JPEG quality 100. For a simple implementation, we put a JPEG encoder at the front of the network and compress non-JPEG images to JPEG images using quality factor 100 with no chroma subsampling.

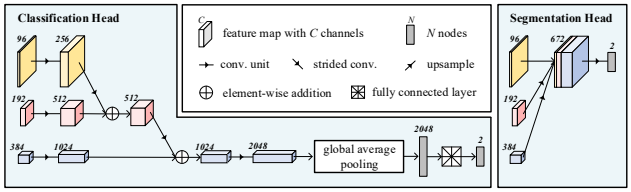


Figure 6. DCT stream classification and segmentation head architecture. Each is attached at the end of the DCT stream to classify double JPEG images for pretraining (Section 3.3) and localize the forgery using the DCT stream only for ablation study (Section 4.4), respectively. RGB stream heads can be similarly constructed using four resolutions.

Method	Input Type	Acc	TPR	TNR
VGG-16 [35]	RGB pixels	50.00	0.00	100.00
Wang [40]	DCT histogram [-5, 5]	73.05	67.74	78.37
Barni [2]	DCT histogram [-60, 60]	84.46	78.35	90.53
Park [26]	DCT histogram [-60, 60] & q. table	92.76	90.90	94.59
DCT stream w/o q. table	DCT volume [-20, 20]	91.71	84.97	97.42
DCT stream (Proposed)	DCT volume [-20, 20] & q. table	93.93	89.43	97.75

Table 1. Double JPEG detection performance (%). The DCT stream, a substream of CAT-Net, showed the highest classification accuracy.

This automatically creates quantized DCT coefficients and a quantization table with all ones.

This is based on the ***compression assumption***: Although a spliced image is saved in uncompressed image format, two source (authentic) images used for the splice forgery were initially compressed in a camera, during acquisition. The file extension for the manipulated image does not matter, *i.e.*, we do not assume a forger saved the forged image in a specific format.

3.3. Pretraining on Double JPEG detection

DCT stream weights are initialized by pretraining on double JPEG detection. The task is to classify whether the given JPEG image has been compressed once or twice. Figure 6 shows that the classification head is attached at the end of the DCT stream since this is a binary classification task. Pretraining on this task helps the stream to capture rich compression artifacts.

We trained and tested the DCT stream on a dataset comprising 1.054M single and double-compressed JPEG images with mixed quality parameters [26]. They compressed raw images from [3, 8, 13] using 1,120 quantization tables including not only 51 standard tables (Q50–Q100) but also non-standard tables obtained from requested images from their public forensic web service. Table 1 shows the double JPEG detection performance of the proposed DCT stream

Dataset		Images	JPEGs	Q. tables	Test
CASIA v2 [10]	auth.	7,491	7,437	50	300
	tamp.	5,105	2,057	7	300
Fantastic Reality [16]	auth.	16,592	16,592	153	1,200
	tamp.	19,423	19,423	1	1,325
IMD2020 [25]	auth.	414	414	58	-
	tamp.	2,010	1,813	73	141
NC16 Splicing [14]	tamp.	288	288	3	288
Carvalho [9]	auth.	100	0	-	100
	tamp.	100	0	-	100
Columbia [24]	auth.	183	0	-	183
	tamp.	180	0	-	180
Spliced COCO (Section 4.1)	tamp.	917,648	917,648	50	4,816

Table 2. Splicing datasets employed in the experiments.

(93.93%), which is the state-of-the-art performance compared with baselines [40, 2, 26]. The proposed network outperformed state-of-the-art neural network [26], which used histogram, even though we used a smaller range of coefficients. Hence, the binary volume representation is a good alternative to the DCT histogram for double JPEG detection.

We also investigated networks without quantization table multiplication to evaluate the effectiveness of using quantization tables. This differed from the original DCT stream in that the quantization table path and concatenation in Figure 5 were removed. Using quantization tables improved double JPEG detection accuracy. Thus, we have adopted quantization tables for forgery localization for the first time.

4. Experiments

4.1. Datasets

Table 2 summarizes the splicing datasets employed in the experiments. We also report the number of Y-channel quantization tables for the first time. Various quantization tables are used, including standard and custom tables, to simulate real-world forgery.

CASIA v2 [10] is a popular dataset for image forgery localization, including images from several sources. We use masks provided by a third-party user [28] since official ground truth masks are not provided. **Fantastic Reality** [16] includes authentic and spliced images for various scenes along with pixel-level ground truth masks. Although authentic images have diverse (153) quantization tables, tampered images have only one quantization table. **IMD2020** [25] includes real-life manipulated images as well as manually created ground truth masks. This dataset contains the most diverse quantization tables because images were collected from the Internet and hence reflects real-world compression schemes. **NC16 Splicing** [14] is a subset of NC16 provided by the National Insti-

tutes of Standards and Technology (NIST). NC16 contains high resolution and challenging manipulated images. Although there are several forgery types, we only use splicing forgery. **Carvalho** [9] DSO-1 contains images of people. Forgeries were created by adding one or more individuals from one to another image with post-processing to increase photorealism. Blocking artifacts are evident when zoomed in, which means that although the images are not in JPEG format, the source images were JPEG compressed, which satisfies the compression assumption (Section 3.2). **Columbia** [24] is a historic dataset for manipulation detection. Ground truth masks are obtained by taking the difference between authentic and forged images followed by some post-processing. The images in this dataset were not compressed in a camera, which violates the compression assumption (Section 3.2).

Quantization tables for tampered images were not diverse, except for IMD2020. Therefore, we created another dataset (**Spliced COCO**) to avoid overfitting specific compression parameters, using COCO 2017 dataset [19] with various quantization tables. Similarly to [25, 45], spliced images were automatically created by selecting one or more arbitrary objects in one image and pasting them onto another image at random positions, with random rotation and resizing. These images were then compressed at random JPEG quality factor 50–99. We did not apply other post-processing, such as blurring the spliced boundary, because that might mislead the network to act like a blur-detector.

We used CASIA v2 (auth./tamp.), Fantastic Reality (auth./tamp.), IMD2020 (tamp.), and Spliced COCO (tamp.) for a training set; and the remaining datasets for testing only. The rightmost column in Table 2 shows the number of images used for testing. We used authentic images too, in contrast with previous image forgery localization studies. We expect this to help the network learn absolute boundaries between tampered and authentic regions, rather than relative boundaries to predict the most suspicious region per image.

4.2. Implementation Details

We initialized the weights of the network by pretraining on ImageNet classification [17] for the RGB stream and double JPEG classification for the DCT stream (Section 3.3). We sampled a balanced number of images in each dataset to construct one epoch, to better handle the high variety of dataset sizes. Training images were cropped to 512×512 patches aligned with an 8×8 grid. Full-resolution images were used for testing, which was possible since the proposed network was fully convolutional.

The network was implemented with PyTorch [27], using stochastic gradient descent with a momentum of 0.9 for the optimizer. The batch size was 24. The learning rate started from 0.005 and decayed exponentially. The objective was to minimize the pixel-wise binary cross entropy loss with fivefold more weight on tampered class. The experiments were performed using 2x NVIDIA TITAN RTX.

4.3. Evaluation Metrics

Our task is a binary segmentation, labeling each pixel in the input image as tampered (positive, 1) or authentic (negative, 0). Thus, each output pixel can be marked as true positive (G:1, P:1), true negative (G:0, P:0), false positive (G:0, P:1), and false negative (G:1, P:0), where G is the ground truth mask and P is the prediction output. G and P are 2-dimensional binary arrays with the same size as the input image.

We evaluated network performances using mean intersection over union (mIoU), a popularly used metric for semantic segmentation [12]. For the two-class case, $\text{mIoU}(G, P) = \frac{1}{2} \cdot \frac{\#(G \cap P)}{\#(G \cup P)} + \frac{1}{2} \cdot \frac{\#(G^{\complement} \cap P^{\complement})}{\#(G^{\complement} \cup P^{\complement})} = \frac{1}{2} \cdot \frac{\text{TP}}{\text{TP+FP+FN}} + \frac{1}{2} \cdot \frac{\text{TN}}{\text{TN+FP+FN}}$, where $\#(\cdot)$ is the number of positive pixels and $^{\complement}$ negates (flips) the mask.

Following [15], we also used the permuted metrics for evaluation. Permuted mIoU is defined as: $\text{p-mIoU}(G, P) = \max\big(\text{mIoU}(G, P), \text{mIoU}(G, P^{\complement})\big)$. For forgery localization tasks, it is sometimes ambiguous which of the two segments is spliced. Permuted metrics measure how well a model can distinguish authentic and tampered regions, rather than its ability to say which is which.

However, mIoU is inappropriate for authentic images, since every pixel in the ground truth mask is negative. Therefore, we used pixel accuracy for testing authentic images: $\text{Acc}(G, P) = \frac{\#(G \cap P) + \#(G^{\complement} \cap P^{\complement})}{\#(G \cup G^{\complement})} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$.

Similarly, permuted pixel accuracy is defined as: $\text{p-Acc}(G, P) = \max\big(\text{Acc}(G, P), \text{Acc}(G, P^{\complement})\big)$. Each metric was calculated per image and averaged over a dataset.

4.4. Results

This section summarizes CAT-Net performance. Tables 3 and 4 show results for test splits and completely unseen images, respectively. We tested ManTra-Net [41] and EXIF consistency [15] to compare CAT-Net with current state-of-the-art image manipulation detectors. Results for those two networks are reported only for completely unseen datasets to ensure fair comparison. We also report performances for the two CAT-Net sub-streams for ablation study and we includes a robustness test for JPEG compression (Figure 7). Figures 8 and 9 show some typical prediction outcomes.

The codes for the two baseline networks were obtained from official public repositories along with their trained weights. ManTra-Net could not test some NC16 Splicing images with full resolution due to GPU memory constraints (NVIDIA TITAN RTX 24GB). Therefore, we cropped those images to QHD (2560×1440) for all networks. A normalized cut was used to aggregate patch-wise predictions

Network	CASIA v2				Fantastic Reality				IMD2020		Spliced COCO	
	authentic		tampered		authentic		tampered		tampered		tampered	
	Acc	p-Acc	mIoU	p-mIoU	Acc	p-Acc	mIoU	p-mIoU	mIoU	p-mIoU	mIoU	p-mIoU
CAT-Net (Proposed)	99.66	99.66	87.63	87.69	99.73	99.75	93.31	93.31	76.00	76.53	93.87	93.87
RGB stream only	99.47	99.70	77.54	77.54	99.89	99.89	92.14	92.14	74.52	74.78	93.80	93.80
DCT stream only	97.83	97.83	83.00	83.12	99.47	99.59	83.91	83.94	68.89	69.64	81.08	81.10

Table 3. Image splicing detection and localization performance for test splits (%).

Network	NC16 Splicing		Carvalho				Columbia			
	tampered		authentic		tampered		authentic		tampered	
	mIoU	p-mIoU	Acc	p-Acc	mIoU	p-mIoU	Acc	p-Acc	mIoU	p-mIoU
CAT-Net (Proposed)	68.41	69.18	99.79	99.79	79.44	79.44	99.54	99.54	83.05	90.09
RGB stream only	60.04	61.25	99.85	99.85	61.17	61.17	99.60	99.60	85.04	89.22
DCT stream only	54.76	59.31	99.33	99.33	78.84	78.84	99.37	99.37	39.34	40.49
ManTra-Net [41]	50.12	50.34	98.65	98.65	56.28	56.46	95.66	95.66	52.34	52.40
EXIF consistency [15]	48.68	53.55	62.04	63.56	48.40	51.33	67.60	68.20	80.81	85.29

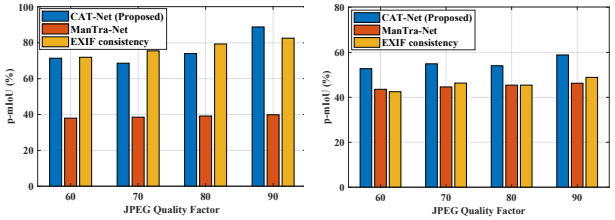
Table 4. Image splicing detection and localization performance for completely unseen datasets (%).

for the EXIF consistency network.

Table 4 and Figure 8 show that CAT-Net excelled in almost all datasets for authentic and tampered images, compared with current state-of-the-art neural networks. The comparison networks always detect some region as tampered, since they are anomaly detectors, even for authentic images. However, CAT-Net produces less false positives since it is a segmentation model and we used authentic images during training. Differences between CAT-Net and the other networks were much larger for tampered images. Thus, CAT-Net was very effective tracking fine traces even if forged images were compressed, *e.g*. NC16 Splicing. Hence, CAT-Net achieved state-of-the-art performance in terms of detecting and localizing real-world image splicing forgeries.

Tables 3, 4, and Figure 9 show that RGB and DCT streams complementary cooperated to improve network performance. For example, in Carvalho (tamp.), the DCT stream performed better; whereas in NC16 Splicing (tamp.), the RGB stream performed better. In both cases, the full network performed best. As discussed in Section 4.1, Columbia violates the compression assumption. Here, the DCT stream couldn't predict well since the images were not compressed at the beginning, leaving no compression artifacts. However, the full network (CAT-Net) performed well on this dataset, with the help of the RGB stream.

Figure 7 shows robustness on JPEG compression tested by compressing Columbia and Carvalho using quality factor 60–90. When additional compression was applied, all three network performances were degraded for Columbia, which had splicing created by two different cameras without compression. In Carvalho, additional compression surely decreased the performance, but the change was smaller because images have initial compression traces, which helped the networks to detect a spliced object. CAT-Net achieved good performance for various quality factors.



(a) Columbia [24] (b) Carvalho [9]

Figure 7. Robustness test on JPEG compression. CAT-Net showed the highest robustness for most of the JPEG quality factors.

5. Conclusion

We have proposed CAT-Net which localizes spliced regions on given images. CAT-Net was the first attempt to consider RGB and DCT domains simultaneously to effectively learn forensic features for visual and compression artifacts remaining in each domain through the RGB and DCT streams. In particular, the DCT stream, containing the JPEG artifact learning module, achieved outstanding performance detecting double JPEG compression. We applied transfer learning from double JPEG detection tasks to image forgery localization tasks for the first time. This helped the network to distinguish statistical fingerprints between spliced and authentic regions. CAT-Net achieved state-of-the-art performance on localizing spliced regions for JPEG or non-JPEG images on diverse datasets compared with current networks.

Acknowledgement This work was supported by Institute of Information & communications Technology Planning & evaluation (IITP) grant funded by the Korea government (MSIT) (2017-0-01671).

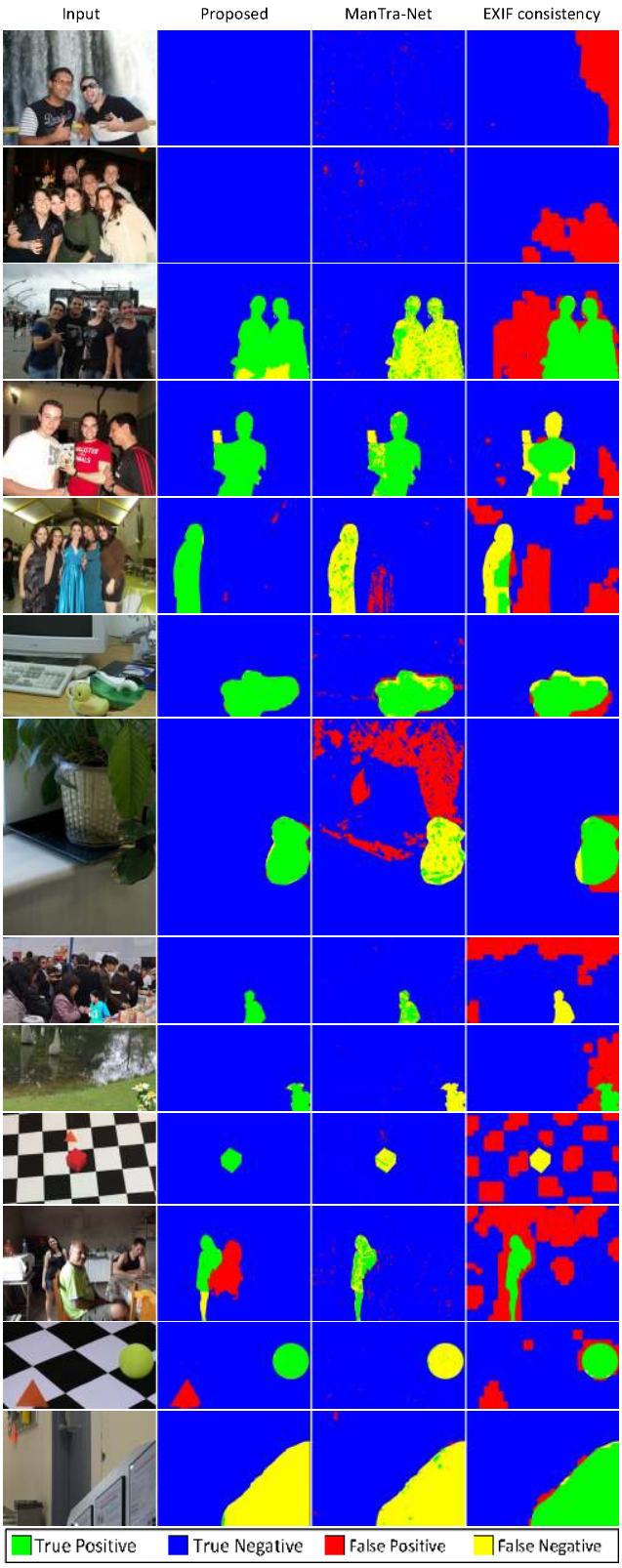


Figure 8. Image splicing localization results for the proposed network and two state-of-the-art networks. Ground truth mask is the union of TP and FN.

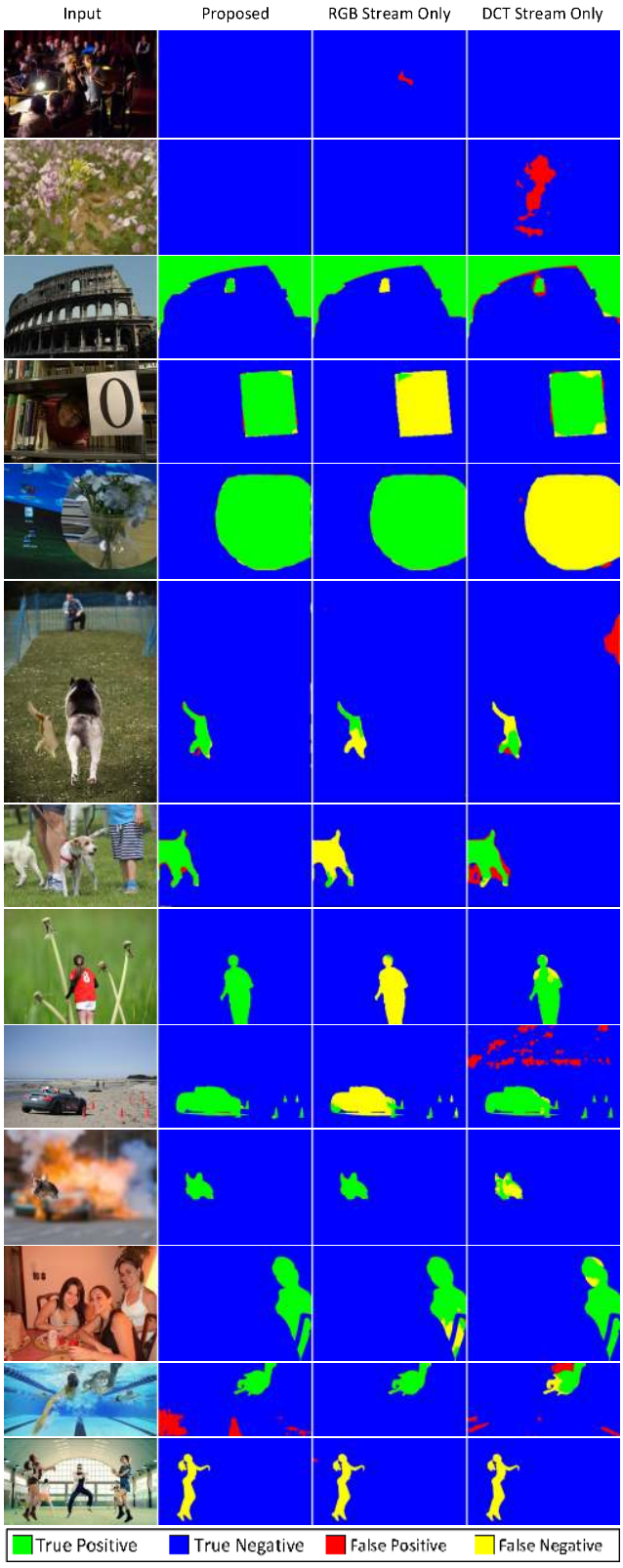


Figure 9. Image splicing localization results for the proposed network and its sub-streams. Ground truth mask is the union of TP and FN.

References

[1] Irene Amerini, Lamberto Ballan, Roberto Caldelli, Alberto Del Bimbo, and Giuseppe Serra. A sift-based forensic method for copy–move attack detection and transformation recovery. *IEEE transactions on information forensics and security*, 6(3):1099–1110, 2011.

[2] Mauro Barni, Luca Bondi, Nicolò Bonettini, Paolo Bestagini, Andrea Costanzo, Marco Maggini, Benedetta Tondi, and Stefano Tubaro. Aligned and non-aligned double jpeg detection using convolutional neural networks. *Journal of Visual Communication and Image Representation*, 49:153–163, 2017.

[3] Patrick Bas, Tomáš Filler, and Tomáš Pevnỳ. " break our steganographic system": the ins and outs of organizing boss. In *International workshop on information hiding*, pages 59–70. Springer, 2011.

[4] Belhassen Bayar and Matthew C Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, 13(11):2691–2706, 2018.

[5] Xiuli Bi, Yang Wei, Bin Xiao, and Weisheng Li. Rru-net: The ringed residual u-net for image splicing forgery detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[6] Mehdi Boroumand, Mo Chen, and Jessica Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, 2018.

[7] Yi-Lei Chen and Chiou-Ting Hsu. Detecting recompression of jpeg images via periodicity analysis of compression artifacts for tampering detection. *IEEE Transactions on Information Forensics and Security*, 6(2):396–406, 2011.

[8] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. Raise: A raw images dataset for digital image forensics. In *Proceedings of the 6th ACM Multimedia Systems Conference*, pages 219–224, 2015.

[9] Tiago José De Carvalho, Christian Riess, Elli Angelopoulou, Helio Pedrini, and Anderson de Rezende Rocha. Exposing digital image forgeries by illumination color classification. *IEEE Transactions on Information Forensics and Security*, 8(7):1182–1194, 2013.

[10] Jing Dong, Wei Wang, and Tieniu Tan. Casia image tampering detection evaluation database. In *2013 IEEE China Summit and International Conference on Signal and Information Processing*, pages 422–426. IEEE, 2013.

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

[12] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.

[13] Thomas Gloe and Rainer Böhme. The'dresden image database'for benchmarking digital image forensics. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1584–1590, 2010.

[14] Haiying Guan, Mark Kozak, Eric Robertson, Yooyoung Lee, Amy N Yates, Andrew Delgado, Daniel Zhou, Timothee Kheyrkhah, Jeff Smith, and Jonathan Fiscus. Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 63–72. IEEE, 2019.

[15] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A Efros. Fighting fake news: Image splice detection via learned self-consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 101–117, 2018.

[16] Vladimir V Kniaz, Vladimir Knyaz, and Fabio Remondino. The point where reality meets fantasy: Mixed adversarial generators for image splice detection. In *Advances in Neural Information Processing Systems*, pages 215–226, 2019.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[20] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[21] Jan Lukáš and Jessica Fridrich. Estimation of primary quantization matrix in double compressed jpeg images. In *Proc. Digital forensic research workshop*, pages 5–8, 2003.

[22] Jan Lukas, Jessica Fridrich, and Miroslav Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214, 2006.

[23] Owen Mayer and Matthew C Stamm. Forensic similarity for digital images. *IEEE Transactions on Information Forensics and Security*, 15:1331–1346, 2019.

[24] Tian-Tsong Ng, Shih-Fu Chang, and Q Sun. A data set of authentic and spliced image blocks. *Columbia University, ADVENT Technical Report 203-2004-3*, 2004.

[25] Adam Novozamsky, Babak Mahdian, and Stanislav Saic. Imd2020: A large-scale annotated dataset tailored for detecting manipulated images. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops*, pages 71–80, 2020.

[26] Jinseok Park, Donghyeon Cho, Wonhyuk Ahn, and Heung-Kyu Lee. Double jpeg detection in mixed jpeg quality factors using deep convolutional neural network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 636–652, 2018.

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming

Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[28] Nam Thanh Pham, Jong-Weon Lee, Goo-Rak Kwon, and Chun-Su Park. Hybrid image-retrieval method for image-splicing validation. *Symmetry*, 11(1):83, 2019.

[29] Alin C Popescu and Hany Farid. Statistical tools for digital forensics. In *international workshop on information hiding*, pages 128–147. Springer, 2004.

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

[31] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[34] Seung-Jin Ryu, Matthias Kirchner, Min-Jeong Lee, and Heung-Kyu Lee. Rotation invariant localization of duplicated image regions based on zernike moments. *IEEE Transactions on Information Forensics and Security*, 8(8):1355–1370, 2013.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[36] Matthew C Stamm and KJ Ray Liu. Forensic detection of image manipulation using statistical intrinsic fingerprints. *IEEE Transactions on Information Forensics and Security*, 5(3):492–506, 2010.

[37] Luisa Verdoliva. Media forensics and deepfakes: an overview. *arXiv preprint arXiv:2001.06564*, 2020.

[38] Vinay Verma, Deepak Singh, and Nitin Khanna. Block-level double jpeg compression detection for image forgery localization. *arXiv preprint arXiv:2003.09393*, 2020.

[39] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[40] Qing Wang and Rong Zhang. Double jpeg compression forensics based on a convolutional neural network. *EURASIP Journal on Information Security*, 2016(1):23, 2016.

[41] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9543–9552, 2019.

[42] Yassine Yousfi and Jessica Fridrich. An intriguing struggle of cnns in jpeg steganalysis and the onehot solution. *IEEE Signal Processing Letters*, 2020.

[43] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[44] Markos Zampoglou, Symeon Papadopoulos, and Yiannis Kompatsiaris. Large-scale evaluation of splicing localization algorithms for web images. *Multimedia Tools and Applications*, 76(4):4801–4834, 2017.

[45] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1053–1061, 2018.