Fighting Fake News: Image Splice Detection via Learned Self-Consistency

Minyoung Huh Andrew Liu Alexei A. Efros Andrew Owens
UC Berkeley
Berkeley Aritifical Intelligence Laboratory (BAIR)
{minyoungg, ahliu, efros, owens}@eecs.berkeley.edu

Paper ID 122

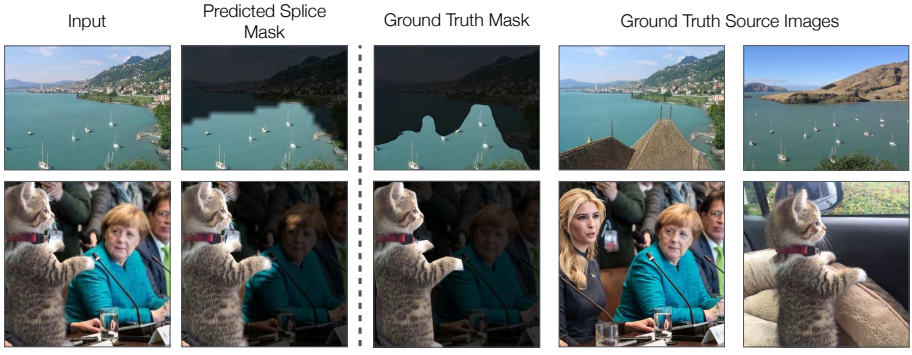

Fig. 1: Our method can detect if an input image has been manipulated and even estimate a splice mask. The method does not require any manipulated images at training time, nor any hard-coded knowledge about image manipulation cues. IMAGE CREDITS: Hays and Efros [1] (top), *Reddit PhotoshopBattles* (bottom).

Abstract. Advances in photo editing and manipulation tools have made it significantly easier to create fake imagery, highlighting the need for better visual forensics algorithms. However, learning to detect manipulations from labelled training data is difficult due to the lack of good datasets for manipulated visual content. In this paper, we introduce a self-supervised method for learning to detect visual manipulations using only unlabeled data. Given a large collection of real photographs with automatically recorded EXIF meta-data, we train a model to determine whether an image is self-consistent – that is, whether its content could have been produced by a single imaging pipeline. We apply this self-supervised learning method to the task of localizing spliced image content. Our forensics model achieves state of the art results on many benchmarks, despite being trained without examples of actual manipulations, and without modeling specific detection cues. Beyond handcrafted benchmarks, we also show promising results spotting fakes on *Reddit* and *The Onion*, as well as detecting computer-generated splices.

Keywords: visual forensics, image splicing, self-supervised learning, EXIF

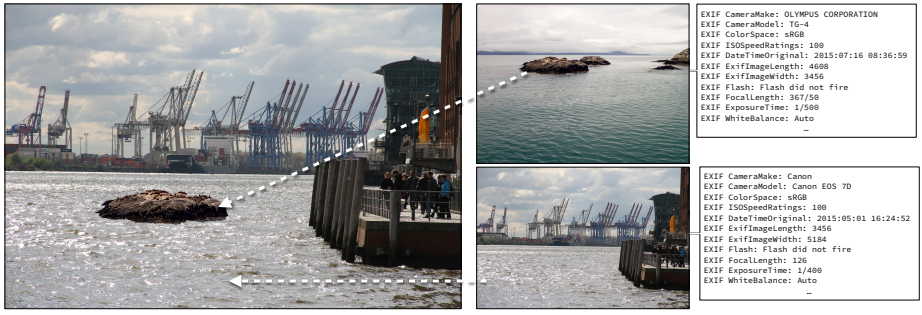

Fig. 2: **Anatomy of a splice:** a fake image is created by splicing together content from two source images. The insight explored in this paper is that patches from spliced images are typically produced by different imaging pipelines, as indicated by the EXIF meta-data of the two source images. The problem is that in practice, we never have access to these source images.

1 Introduction

Malicious image manipulation, long the domain of dictators and spy agencies, has now become accessible to legions of common Internet trolls and Facebook con-men [2]. With only rudimentary Photoshop skills, it is now possible to create realistic image composites [3, 4], fill in large image regions [1, 5, 6], generate plausible video from speech [7], etc. One might have hoped that these new methods for creating synthetic visual content would be met with commensurately powerful techniques for detecting fakes, but this has not been the case so far.

The main problem is that standard supervised learning approaches, which have been very successful for many types of detection problems, are not well-suited for visual image forensics. This is because the space of manipulated images is so vast and diverse, that it is rather unlikely we will ever have enough manipulated training data for a supervised method to fully succeed. Indeed, detecting visual manipulation can be thought of as an anomaly detection problem – we want to flag anything that is "out of the ordinary", even though we might not have a good model of what that might be. In other words, we would like a method that does not require any manipulated training data at all, but can work in an unsupervised/self-supervised regime.

In this work, we turn to a vast and previously untapped source of data, image EXIF meta-data. EXIF tags are camera specifications that are digitally engraved into an image file at the moment of capture and are ubiquitously available. Consider the photo shown in Figure 2. While at first glance it might seem authentic, we see on closer inspection that a small island has been inserted into the water. The content for this spliced region came from a different photo, shown on the right. Such a manipulation is called an *image splice*, and it is the most widely-used way of creating visual fakes. If we had access to the two source photographs, we would see from their EXIF meta-data that there are a number of important differences in the imaging pipelines: one photo was taken with an *Olympus* camera, the other with a *Canon* camera; the images were shot using different focal lengths, and saved with different JPEG quality settings, etc. Our insight is that one

might be able to recognized spliced images because they are are made up of regions that were captured with different imaging pipelines. Of course, in forensics applications, we do not have access to the original source images nor, in general, do we even have access to the fraudulent photo's meta-data. This poses a challenge when trying to design methods that can leverage EXIF cues.

We propose a self-supervised method for identifying and localizing image splices by predicting the consistency of EXIF attributes between pairs of patches to determine whether they came from a single coherent image. We validate our approach using several benchmark datasets, and show that the model performs better than the state-of-the-art — despite never seeing annotated splices or using hand-engineered cues.

2 Related work

Over the years, researchers have proposed a variety of visual forensics methods for identifying various image manipulations. Early work relied on domain knowledge to isolate physical cues within an image. Drawing upon techniques from signal processing, previous methods focused on cues such as misaligned JPEG blocks [2], compression quantization artifacts [8], resampling artifacts [9], color filtering array discrepancies [10], and camera-hardware "fingerprints" [11]. We take inspiration from the recent work of Agarwal and Farid [12], which exploits a seemingly insignificant differences, such as how different cameras truncate numbers during JPEG quantization, to detect spliced image regions. While these domain-specific approaches have proven to be extremely useful, particularly due to their easy interpretability, we believe that the use of machine learning should open the door to discovering many more useful cues, while also producing more adaptable algorithms.

Indeed, recent work has moved away from using *a priori* knowledge and toward applying end-to-end learning methods for solving specific forensics tasks such as distinguishing camera models [13] and finding double JPEG compression [14]. This line of work has made remarkable progress in training forensics algorithms without using specifically-crafted features and algorithms.

In our work, we seek to further reduce the amount of information we provide to the algorithm by having it learn to detect manipulations without the need for human-labelled data. For this, we take inspiration from recent works in self-supervision [15, 16, 17, 18, 19, 20], which train models by solving tasks solely defined using unlabeled data. Of these, the most closely related approach is perhaps that of Doersch et al. [16], which trained a model to predict the relative position of pairs of patches within an image. Surprisingly, the authors found that their method learned to detect and utilize very subtle artifacts like chromatic lens aberration, as a short-cut for learning the task. While camera information was a nuisance signal in their work, it is a useful signal for us. Our approach is also closely related to Joulin [21], which predicted meta-data from large numbers of Flickr photos. However, this meta-data was semantic, rather than the low-level camera meta-data that our model predicts.

Our work is also related to the anomaly detection problem. But unlike traditional visual anomaly detection work, which is largely concerned with detecting unusual semantic events, such as the presence of rare objects and actions [22], here we are inter-

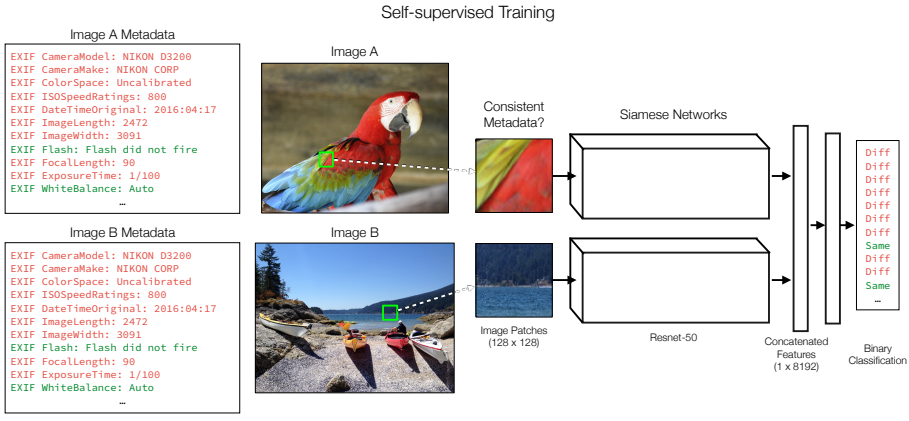

Fig. 3: **Self-supervised Training:** Our model takes two random patches from two different images and predicts whether or not they have consistent meta-data. Each attribute is used as a consistency metric during training.

ested in finding anomalies in photos whose content is plausible enough to fool humans. The anomalies we search for should therefore be mostly imperceptible to humans, and invariant to the semantics of the scene.

3 Learning Image Self-consistency

Our model works by predicting whether pairs of patches from the same image are consistent with each other. Given two patches, $\mathcal{P}_i$ and $\mathcal{P}_j$, we estimate the probabilities $x_1, x_2, ..., x_n$ that they share the same value for each of n EXIF attributes. We then estimate the patches' consistency, c_{ij}, by combining our n observations of EXIF attribute consistency. Finally, we compute an estimate of image self-consistency by aggregating a set of pairwise patch consistency probabilities into global heat-map. Upon acceptance, we will release our model, code, and data online.

3.1 Predicting EXIF Attribute Consistency

We use a Siamese network to predict the probability that a pair of 128×128 image patches share the same value for each EXIF attribute. We train this network with image patches randomly sampled from $400K$ untampered Flickr photos, predicting all EXIF attributes that appeared in more than $50K$ photos ($n = 80$, the full list of attributes can be found in the supplementary material).

In practice, training using this simple random sampling procedure will not be successful because: 1) there are some rare EXIF values that will be very difficult to learn, and 2) randomly selected pairs of images are unlikely to have consistent EXIF values just by chance. Therefore, we introduce two types of re-balancing during training: unary and pairwise. For unary re-balancing, we oversample rare EXIF attribute values

(e.g. rare camera models). When constructing a mini-batch, we first choose an EXIF attribute and uniformly sample an EXIF value from all possible values of this attribute. For pairwise re-balancing, we make sure that pairs of training images within a mini-batch are selected such that for a given EXIF attribute, half the batch share that value, and half do not.

Many image manipulations are performed with the intent to make the resulting image look plausible to the human eye: spliced regions are resized, edge artifacts are smoothed, and the resulting image is re-JPEGed. These operations can disrupt the low-level camera signal that was once present in the original images. To mimic these operations, we add three augmentation operations during training: re-JPEG, Gaussian blur, and image resize. During training, with probability p, we apply one or more of these augmentation operations to both patches. The parameters of each operation are chosen randomly; but half of the time, both patches use the same parameters, and half of the time, different. We then introduce three corresponding classification labels which are used to train the model to predict whether a pair of patches had received the same parameterized augmentation or not. This increases our 80-way classification to 83-way. Since the order of these operations matter, they are applied in a random order during batching.

3.2 Predicting Patch Consistency

Once we have predicted the consistency of a pair of patches for each of our EXIF plus differential attributes, we would like to estimate the pairs' *overall* consistency c_{ij}. If we were solving a supervised task, then a natural choice would be to use spliced regions as supervision to regress from n EXIF consistency predictions to the probability that the two patches belong to different splices. Unfortunately we do not have spliced images to train on. Instead, we use a self-supervised proxy task: we train a four-layer classifier to predict whether the two patches come from *the same image* or not. Note that this is similar to an unsupervised anomaly detection model.

3.3 From Patch Consistency to Image Self-Consistency

So far we have introduced a model which can measure some form of consistency between pairs of patches within an image. In order to transform this into something usable for detecting splices, we need to aggregate these pairwise consistency probabilities into a global self-consistency score for the entire image.

Given an image, we sample approximately 500 patches in a grid - selecting a stride dependent on the size of the image - and construct an affinity matrix by computing consistencies between every pair of patches. That is, for a given patch, we can visualize a response map corresponding to its consistency with every other patch in the image. To increase the spatial resolution of each response map, we average the predictions of overlapping patches. If there is a splice, then the majority of natural patches will have low consistency with patches from the manipulated region while having high consistency with each other. (see Figure 4c).

To produce a single response map for the image, we want to find the most consistent mode among all the 500 patch response maps. We do this mode-seeking using Mean

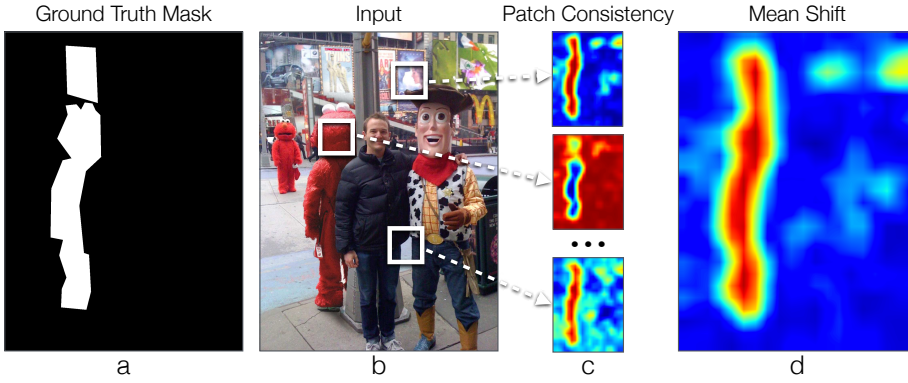

Fig. 4: **Test Time:** Our model samples patches in a grid from an input image (b) and estimates consistency for every pair of patches. (c) For a given patch, we get a consistency map over all other patches in the image. (d) We use Mean Shift to aggregate the consistency maps into a final prediction.

Shift [23]. The resulting response map naturally segments the image into consistent and inconsistent regions (see Figure 4d).

4 Results

We evaluate our self-consistency model on two closely related tasks: splice detection and splice localization. In the former, our goal is to simply classify images as being spliced *vs.* authentic. In the latter, the goal is to localize the spliced regions within an image.

4.1 Benchmarks

We evaluate our method on 5 different benchmarks, three existing visual forensics benchmarks and two new benchmarks. These existing benchmarks are: the widely-used *Columbia* dataset [24], which consists of 180 relatively simple splices, and two more challenging datasets, *Carvalho* et al. [25] (94 images) and *Realistic Tampering* [26] (220 images), which combine splicing with post-processing operations to create images that appear more realistic. In addition, we introduce a new *In-the-Wild* dataset, and a dataset of automatically-generated spliced imagery.

One potential shortcoming of these existing benchmarks is that they may not be representative of the wide variety of forgeries encountered online. This is because the benchmarks have been created by a small number of artists, biasing the data. Hence, we introduce a new forensics benchmark called *In-the-Wild* that consists of 201 images scraped from THE ONION, a parody (i.e. fake news!) news website, and REDDIT PHOTOSHOPBATTLES, a community-driven dataset, where each user attempts to photoshop a given image. By finding real-world examples of image forgeries, we gain authenticity

Dataset	**Columbia [24]**	**Carvalho [25]**	**RT [26]**
CFA [27]	0.83	0.64	0.54
DCT [28]	0.58	0.63	0.52
NOI [29]	0.73	0.66	0.52
Supervised FCN	0.57	0.56	**0.56**
Camera Classification	0.70	0.73	0.15
Same-image	0.88	0.72	0.55
Self-Consistency	**0.93**	**0.77**	**0.56**

Table 1: **Splice Detection.** We compare our results on 3 different benchmarks. These benchmarks consist of both spliced and authentic images, which allows us to compute the mean average precision (mAP) on whether an image has been spliced.

at the cost of losing rigorous ground truth. We labeled our dataset by hand, identifying the most obvious splices within each image. Having access to the source image aided us in the labeling, however, we cannot guarantee the 100% correctness of our labels.

Furthermore, we also want to evaluate our method on automatically-generated splices. For this, we used the scene completion benchmark from [1], which comes with inpainting results, masks, and source images for a total of 55 splices. We note that the masks are an approximation to the true splices since the scene completion algorithm modifies a small region of pixels outside the mask in order to produce seamless splices.

4.2 Comparisons

We compare against three baseline algorithms, all of which use classic image processing to explicitly model and exploit imaging artifacts: Color Filter Array (CFA) [27] detects artifacts in color pattern interpolation; JPEG DCT [28] detects inconsistencies over JPEG coefficients; and Noise Variance (NOI) [29] detects anomalous noise patterns using wavelets. We used implementations of these algorithms provided by [30].

Since we also wanted to compare our unsupervised method with supervised methods, we also provide a comparison against a concurrent, unpublished learning-based method E-MFCN [31]. Given a dataset of spliced images and masks as training data, they employ supervised fully-convolutional networks (FCN) [32] to predict splice masks as well as splice boundaries in test images. As a supervised method, their model requires a large training set of splices that we were unable to obtain from the authors. Therefore, we only report E-MFCN performance on the 3 standard datasets. To evaluate performance on the other datasets, we have implemented a simplified version of the algorithm using a standard FCN, and trained it on portions of the *Columbia*, *Carvalho*, and *Realistic Tampering* datasets.

Finally, we also present two simplified versions of our full *Self-Consistency* model. The first, (*Camera Classification*), is trained to directly predict which camera model produced a given image patch. We evaluate the output of the camera classification model, by sampling images patches from a test image and assigning the most frequently predicted camera model as one class and everything else to be the other. Here, an image

	Columbia [24]			Carvalho [25]			RT [26]	In-the-Wild	Hays [1]
Model	MCC	F1	mAP	MCC	F1	mAP	mAP	mAP	mAP
CFA [27]	0.23	0.47	0.84	0.16	0.29	0.58	**0.69**	0.56	0.58
DCT [28]	0.33	0.52	0.58	0.19	0.31	0.60	0.53	0.55	0.52
NOI [29]	0.41	0.57	0.69	0.25	0.34	0.66	0.58	0.65	0.60
E-MFCN [31]	0.48	**0.61**	-	**0.41**	0.48	-	-	-	-
Supervised FCN	-	-	0.77	-	-	0.59	0.58	0.55	0.57
Camera Classification	0.29	0.51	0.59	0.15	0.55	0.53	0.51	0.54	0.55
Same Image	0.45	0.56	0.87	0.23	0.72	0.64	0.58	0.66	**0.66**
Self-Consistency	**0.54**	**0.61**	**0.89**	0.32	**0.76**	**0.72**	0.58	**0.70**	**0.66**

Table 2: **Splice Localization**. We compare our results against 5 different datasets using mean average precision (mAP) on the spliced images. MCC and F1 are per-pixel metric that is computed after thresholding. We report these numbers as they were the only numbers reported by [31].

is considered completely untampered when every patch's predicted camera model is the same.

The second model, *Same Image*, is a Siamese network that directly predicts whether two patches are sampled from the same image. An image is considered likely to be tampered if its constituent patches are predicted to have come from different images. The evaluations of the simplified models are performed the same way as in our full *Self-Consistency* model.

4.3 Splice Detection

We evaluate splice detection using three datasets, *Columbia*, *Carvalho*, and *Realistic Tampering*, that contain both untampered and manipulated images. For each algorithm, we extract the output map and obtain an overall score by averaging the output pixels. The images are ranked based on their overall scores and we compute the mean average precision(mAP) for the whole dataset. We chose to use mAP as it is a condensed representation of the precision-recall curves previously used by [30, 27, 28, 29].

Table 1 shows the mAP for detecting manipulated images. Our Self-Consistency model achieves state-of-the-art performance on *Columbia* and *Carvalho* while producing results comparable to supervised *FCN* on Realistic Tampering.

4.4 Splice Localization

Having seen that our model can distinguish spliced and authentic images, we next ask whether it can also localize spliced regions within images. To evaluate our performance, we use a mean average precision (mAP) metric. More specifically, for each image in our dataset, our algorithm produces an unnormalized probability that each pixel was part of a splice. We evaluate the quality of this prediction by comparing it with the ground-truth mask using average precision (AP), and take the mean of these AP estimates over the test set to get an overall score. Since there is ambiguity in which of the two regions is the splice and which is the original, we compute AP for both permutations and take the

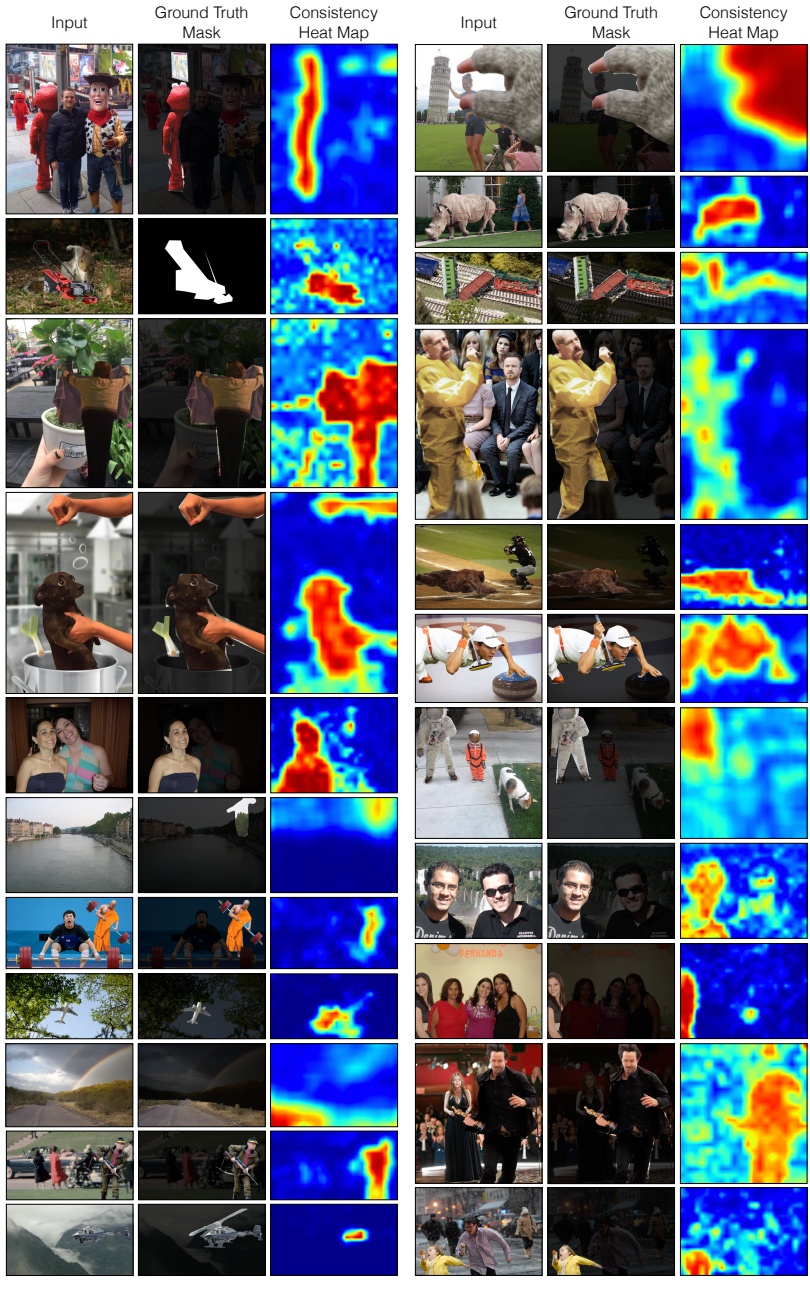

Fig. 5: **Detecting Fakes:** *Self-Consistency* successfully localizes manipulations across many different datasets. We show qualitative results on images from *Carvalho*, *In-the-Wild*, *Hays* and *Realistic Tampering*.

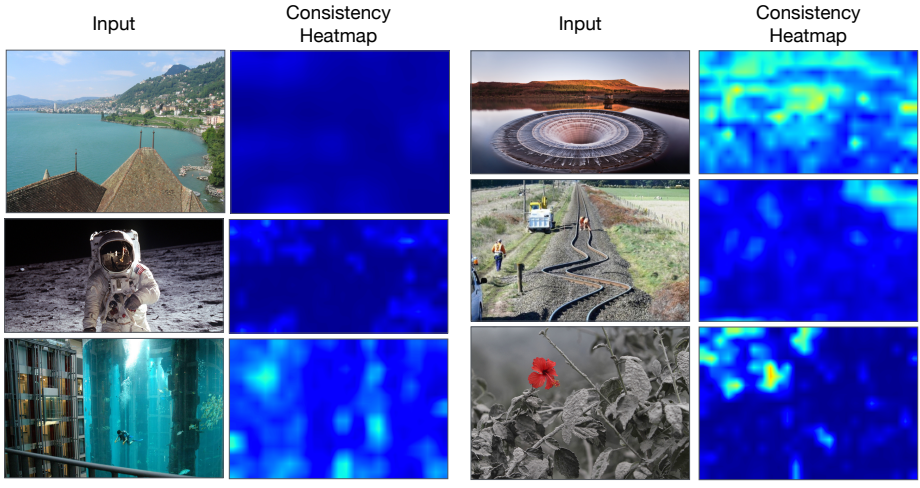

Fig. 6: **Response on real images:** It's important to note that our method is generally not fooled by untampered images.

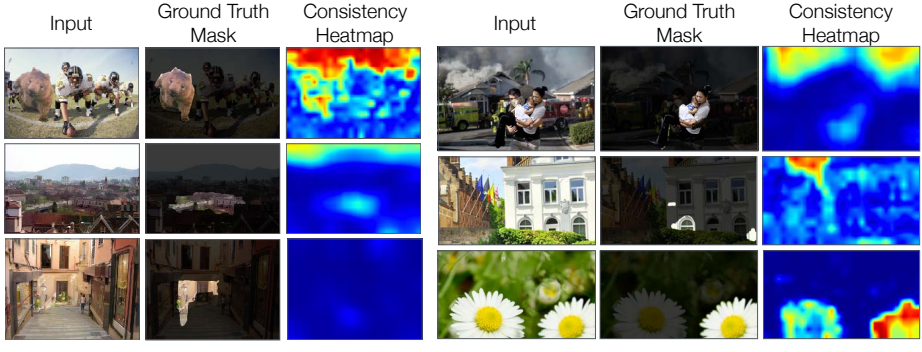

Fig. 7: **Failure Cases:** We present typical failure modes of our model. As we can see with outdoor images, overexposure frequently leads to false positives in the sky. In addition some splices are too small that we cannot effectively locate them using consistency. Finally the flower copy-move confuses *Self-Consistency* because inconsistent differential augmentation will disagree with consistent EXIF values.

higher one (we do this for all methods). We also report our results on metrics such as MCC and F1, so that we could directly compare with previous forensics work [31].

The quantitative results on Table 2 show that our *Self-Consistency* model achieves the best performance across all datasets with the exception of the *Realistic Tampering (RT)* dataset. Notably, the model generally outperformed the supervised baselines, which was trained with actual manipulated images, despite the fact that our model never saw a tampered image at training time. We suspect that the supervised models' poor per-

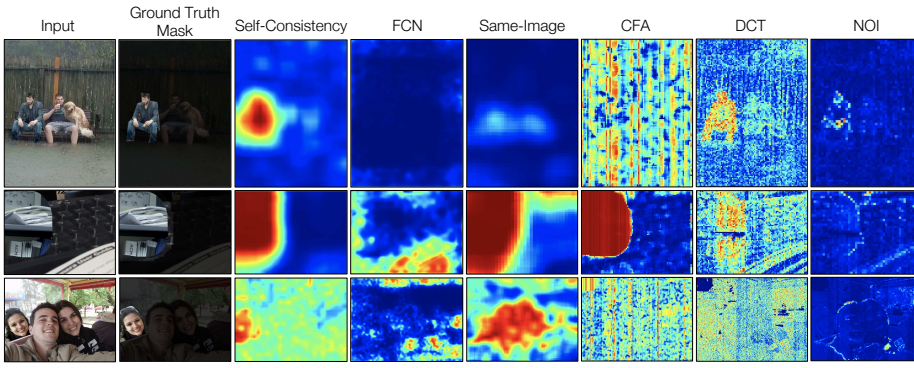

Fig. 8: **Competing Methods:** We visualize the qualitative difference between *Self-Consistency* and baselines. Our model can correctly localizes image splices from *In-the-Wild*, *Columbia* and *Carvalho* that other methods struggle on.

formance may be due to the difficulty in generalizing to manipulations made by artists whose work was not present at training time.

We also observed that our self-consistency model outperforms our *Same Image* method that was trained to predict whether pairs of patches come from the same image. We suspect that this may be due to bias during training. At training time, it suffices to solve the self-supervised learning task by matching the patches' general appearance (e.g., color histograms). At test time, however, a forger will have intentionally concealed such obvious cues.

It is also instructive to look at the qualitative results of our method, which we show on Figure 5. Observe that our method appears to be robust to a wide range of different types of spices, across all datasets. Furthermore, in Figure 6, we show that our method is not fooled by real images, even if they look like they might be photoshopped. We can also look at the qualitative differences between our method and the baselines in Figure 8.

Failure cases In Figure 7 we show some common failure cases. Our performance on *Realistic Tampering* illustrates some shortcomings with *Self-Consistency*. First, our model is not well-suited at finding small splices that are common in *RT*. When spliced regions are small, large striding will skip over the spliced region, mistakenly suggesting that no manipulations exist. Second, over- under-exposure is often flagged by our model to be inconsistent because they lack a meta-data signal. Finally, *RT* contains a significant number of additional manipulations, such as copy-move, that cannot be consistently detected via meta-data consistency, since the manipulated content comes from exactly the same photo.

4.5 Detection cues

One main contribution of our method is the use of EXIF metadata as a proxy task for learning about natural images. To see how useful these 80 EXIF attribute and 3

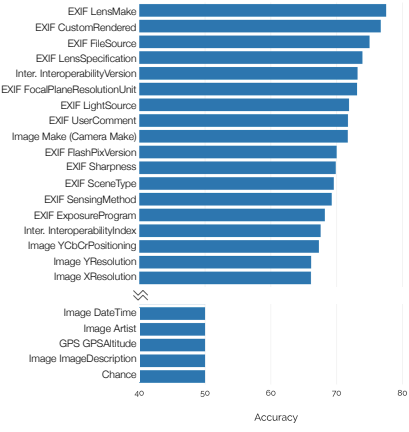

Fig. 9: **EXIF Accuracy** How predictable are EXIF attributes? We compute individual pairwise-consistency accuracy on Flikr images using our self-consistency model.

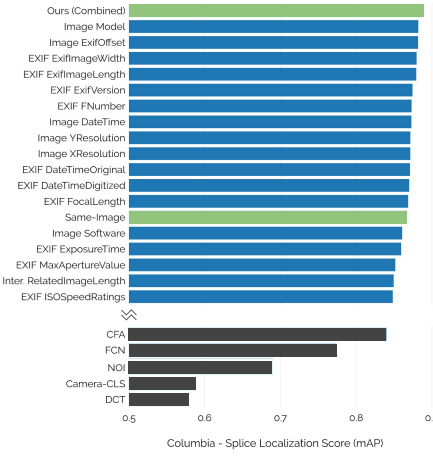

Fig. 10: **EXIF Splice Localization** How useful are EXIF attributes for localizing splices? We compute individual localization score on Columbia dataset.

differential augmentations are, we evaluate their performance in isolation as a splice-localization method. We first investigate the predictive strength of each EXIF attribute by computing the consistency prediction accuracy on 50K natural *Flickr* images. In Figure 9, we order these attributes from best to worst. Since we cannot run our model on all possible image pairs (50000^2 combinations), we compare against a subset.

Unsurprisingly, we have found attributes such as *Image DateTime* and *Image Artist* – attributes that are almost impossible to predict from image-data alone – performed poorly. On the other hand, attributes that are known to leave imaging artifacts (e.g *EXIF LensMake* and software-related attributes) act as very informative consistency cues.

Our self-supervised task is not particularly meaningful without context, so we also show in Figure 10 how useful each EXIF tag is at localizing splices. We believe the differences in the predictability and localization score rankings are due to the biases present in *Columbia* dataset. During training, self-consistency model encounters more than 3000 unique camera models (one benefit of being unsupervised), yet *Columbia* is constructed only using four cameras (a subset of the 3000 camera models seen). This same limitation is also present in *Carvalho* and *RT*. Across the three forensics datasets, only a total of ten unique cameras were used to take pictures for splicing. In Figure 11, we also visualize the individual responses ranked by their splice localization score.

5 Conclusion

In this paper, we proposed a new method for detecting photo manipulations. We introduced a self-supervised technique based on learning to find internal inconsistencies in images: for example identifying photos that could not have been taken by a single

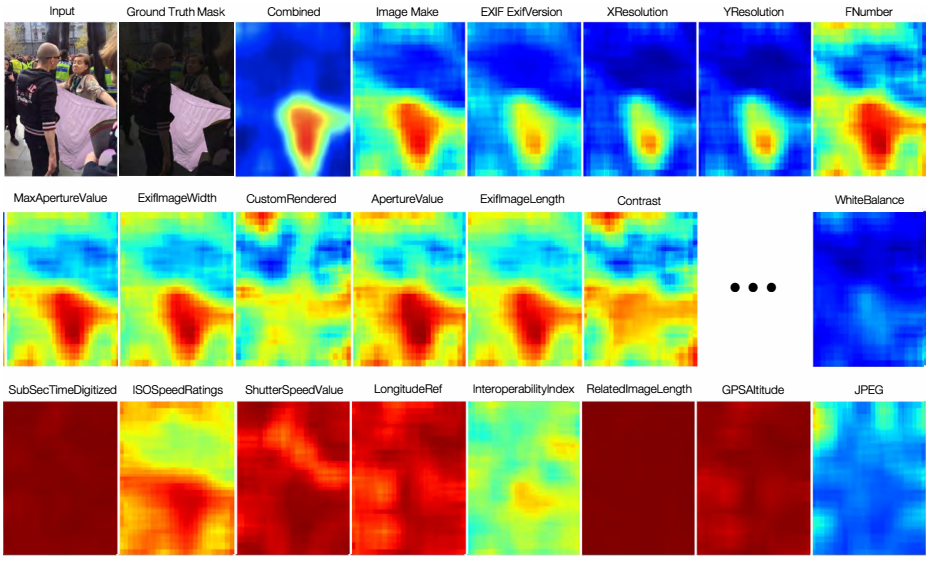

Fig. 11: **Individual EXIF Consistency:** We compute EXIF consistency responses for each attribute and visualize the top and bottom performing based on their localization score.

camera or compressed by a single imaging pipeline. As shown, our method obtains state-of-the-art results across several datasets.

We see this work as opening two directions in visual forensics research. First, we contribute new strategies for using self-supervision to detect manipulations, such as using EXIF meta-data as relative supervisory signal, and creative new supervisory signals by simulating differential manipulations (e.g JPEG, blur, resizing). However, there are many other possible ways to formulate self-supervision tasks that could be used to detect inconsistencies.

The second direction relates to our success in using consistency for detecting anomalies. Consistency is a type of meta-supervision which can exploit new sources of structured information by organizing lower-level cues like EXIF meta-data into higher-level notions that can be used for meaningful higher-level tasks like detecting image manipulations.

References

1. Hays, J., Efros, A.A.: Scene completion using millions of photographs. In: ACM Transactions on Graphics (TOG). Volume 26., ACM (2007) 4 1, 2, 7, 8
2. Farid, H.: Photo forensics. MIT Press (2016) 2, 3
3. Zhu, J.Y., Krahenbuhl, P., Shechtman, E., Efros, A.A.: Learning a discriminative model for the perception of realism in composite images. In: The IEEE International Conference on Computer Vision (ICCV). (December 2015) 2
4. Tsai, Y.H., Shen, X., Lin, Z., Sunkavalli, K., Lu, X., Yang, M.H.: Deep image harmonization. In: CVPR. (2017) 2

5. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Trans. Graph. **28**(3) (2009) 24–1 2
6. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2016) 2
7. Suwajanakorn, S., Seitz, S.M., Kemelmacher-Shlizerman, I.: Synthesizing obama: learning lip sync from audio. ACM Transactions on Graphics (TOG) **36**(4) (2017) 95 2
8. Luo, W., Huang, J., Qiu, G.: Jpeg error analysis and its applications to digital image forensics. IEEE Transactions on Information Forensics and Security **5**(3) (2010) 480–491 3
9. Huang, F., Huang, J., Shi, Y.Q.: Detecting double jpeg compression with the same quantization matrix. IEEE Transactions on Information Forensics and Security **5**(4) (2010) 848–856 3
10. Popescu, A.C., Farid, H.: Exposing digital forgeries by detecting traces of resampling. IEEE Transactions on signal processing **53**(2) (2005) 758–767 3
11. Swaminathan, A., Wu, M., Liu, K.R.: Digital image forensics via intrinsic fingerprints. **3**(1) (2008) 101–117 3
12. Agarwal, S., Farid, H.: Photo forensics from jpeg dimples. Workshop on Image Forensics and Security (2017) 3
13. Bondi, L., Baroffio, L., Gera, D., Bestagini, P., Delp, E.J., Tubaro, S.: First steps toward camera model identification with convolutional neural networks. IEEE Signal Processing Letters **24**(3) (March 2017) 259–263 3
14. Barni, M., Bondi, L., Bonettini, N., Bestagini, P., Costanzo, A., Maggini, M., Tondi, B., Tubaro, S.: Aligned and non-aligned double JPEG detection using convolutional neural networks. CoRR **abs/1708.00930** (2017) 3
15. de Sa, V.: Learning classification with unlabeled data. In: Neural Information Processing Systems. (1994) 3
16. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. ICCV (2015) 3
17. Jayaraman, D., Grauman, K.: Learning image representations tied to ego-motion. In: ICCV. (December 2015) 3
18. Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. In: ICCV. (2015) 3
19. Owens, A., Wu, J., McDermott, J.H., Freeman, W.T., Torralba, A.: Ambient sound provides supervision for visual learning. (2016) 3
20. Zhang, R., Isola, P., Efros, A.A.: Split-brain autoencoders: Unsupervised learning by cross-channel prediction. (2017) 3
21. Joulin, A., van der Maaten, L., Jabri, A., Vasilache, N.: Learning visual features from large weakly supervised data. In: European Conference on Computer Vision, Springer (2016) 67–84 3
22. Hoai, M., De la Torre, F.: Max-margin early event detectors. International Journal of Computer Vision **107**(2) (2014) 191–202 3
23. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence **17**(8) (Aug 1995) 790–799 6
24. Ng, T.T., Chang, S.F.: A data set of authentic and spliced image blocks. (2004) 6, 7, 8
25. d. Carvalho, T.J., Riess, C., Angelopoulou, E., Pedrini, H., d. R. Rocha, A.: Exposing digital image forgeries by illumination color classification. IEEE Transactions on Information Forensics and Security **8**(7) (July 2013) 1182–1194 6, 7, 8
26. Korus, P., Huang, J.: Evaluation of random field models in multi-modal unsupervised tampering localization. In: Proc. of IEEE Int. Workshop on Inf. Forensics and Security. (2016) 6, 7, 8

27. Ferrara, P., Bianchi, T., Rosa, A.D., Piva, A.: Image forgery localization via fine-grained analysis of cfa artifacts. IEEE Trans. Information Forensics and Security **7**(5) (2012) 1566–1577 7, 8
28. Ye, S., Sun, Q., Chang, E.C.: Detecting digital image forgeries by measuring inconsistencies of blocking artifact. In: ICME07. (2017) 7, 8
29. Mahdian, B., Saic, S.: Using noise inconsistencies for blind image forensics. In: IVC09. (2009) 7, 8
30. Zampoglou, M., Papadopoulos, S., Kompatsiaris, Y., Bouwmeester, R., Spangenberg, J.: Web and social media image forensics for news professionals. In: Social Media In the Newsroom, SMNews16@CWSM, Tenth International AAAI Conference on Web and Social Media workshops. (2016) 7, 8
31. Salloum, R., Ren, Y., Kuo, C.J.: Image splicing localization using A multi-task fully convolutional network (MFCN). CoRR **abs/1709.02016** (2017) 7, 8, 10
32. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. CoRR **abs/1605.06211** (2016) 7