2018 IEEE Conference on Multimedia Information Processing and Retrieval

# Detection of GAN-generated Fake Images over Social Networks

Francesco Marra, Diego Gragnaniello, Davide Cozzolino, Luisa Verdoliva
*Dep. of Electrical Engineering and Information Technology*
*University Federico II of Naples*
*Naples, Italy*
{*name.surname*}*@unina.it*

***Abstract*—The diffusion of fake images and videos on social networks is a fast growing problem. Commercial media editing tools allow anyone to remove, add, or clone people and objects, to generate fake images. Many techniques have been proposed to detect such conventional fakes, but new attacks emerge by the day. Image-to-image translation, based on generative adversarial networks (GANs), appears as one of the most dangerous, as it allows one to modify context and semantics of images in a very realistic way. In this paper, we study the performance of several image forgery detectors against image-to-image translation, both in ideal conditions, and in the presence of compression, routinely performed upon uploading on social networks. The study, carried out on a dataset of 36302 images, shows that detection accuracies up to 95% can be achieved by both conventional and deep learning detectors, but only the latter keep providing a high accuracy, up to 89%, on compressed data.**

***Keywords*-image forensics; GAN; convolutional neural networks.**

## I. Introduction

With the capillary diffusion of social networks, spreading information (and doctored information) has become very easy. Fake news are often supported by multimedia contents, aimed at increasing their credibility. Indeed, by using powerful image editing tools, such as Photoshop or GIMP, even non-expert users can easily modify an image obtaining realistic results, which evade the scrutiny of human observers. A study recently published in Cognitive Research [1] tried to measure people's ability to recognize, by visual inspection, whether a photo had been doctored or not. Only 62%-66% of the photos were correctly classified, and users proved even worse at *localizing* the manipulation. In a similar study [2] only 58% of the images were correctly classified, and only 46% of the manipulated images were identified as such. The threat represented by widespread image forgery has stimulated intense research in multimedia forensics. As a result, automatic algorithms, under suitable hypotheses, achieve a much better detection performance than humans. For example, in the first IEEE Image Forensics Challenge, detection accuracies beyond 90% were obtained by means of a machine learning approach with a properly trained classifier [3].

The situation is almost reversed for what concerns computer generated fake images. A recent study [4] proved that, for such images, human judgement is significantly better than machine learning. This may be attributed to the inability of current computer graphics tools to provide a good level of photorealism. As a consequence, the research activity in this field has been less intense. Some papers propose to detect computer graphics images based on statistics extracted from their wavelet decomposition [5], or from residual images [6]. Other papers rely on the different noise introduced by the recording device [7], on traces of chromatic aberrations [8], or traces of demosaicing filters [9]. Differences in color distribution are explored in [10], while in [11] the statistical properties of local edge patches are used for discrimination. In [12] face asymmetry is proposed as a discriminative feature to tell apart computer generated from natural faces. Only very recently, deep learning has been used for this task [4], [13], [14], and found to outperform preceding approaches.

Figure 1. Spot the fake. Two satellite images, one downloaded from Google Maps, the other artificially generated.

So, when fake images are easily detected by humans, the need for sophisticated detectors is less impelling. However, computer graphics technology progresses at a fast pace, and observers find more and more difficult to distinguish between computer-generated and photographic images [15]. Indeed, new forms of image manipulation based on computer graphics have been recently devised, characterized by a much higher level of photorealism [16]–[21]. In particular, we focus, here, on image-to-image translation, a process that modifies the attributes of the target image by *translating* one possible representation of a scene into another one. This attack has become relatively easy, and very popular, with the advent of generative adversarial networks (GANs), whose

0-7695-6354-6/18/$31.00 ©2018 IEEE
DOI 10.1109/MIPR.2018.00084

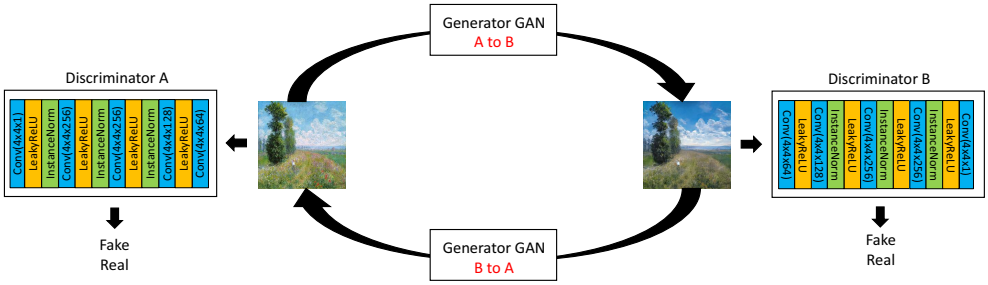


Figure 2. Cycle-GAN based image-to-image translation.

very aim is to generate images indistinguishable from the real images drawn from a given source. Fig.1 provides an example of image-to-image translation.

In this work we analyze the performance of a number of learning-based methods in the detection of image-to-image translation. Our goal is to understand if, to which extent, and in which conditions, these attacks can be unveiled. To this end we will consider several solutions, based both on state-of-the-art methods taken from the image forensic literature, and on general purpose very deep convolutional neural networks (CNNs) properly trained for this task. We will also study the case in which images are posted over a social network, like Twitter. In fact, this is both the most common and most challenging situation, since the compression routinely performed upon image uploading tends to impair the performance of forgery detectors [22].

In the following Sections, we will describe the generation process of the images analyzed in the paper, referring to the method proposed in [18], present the approaches used for detection, and finally discuss the experimental results.

## II. GAN-based Image-to-Image Translation

Image-to-image translation is a very general problem wherein images of some domain are mapped to corresponding images in another domain. Recently, GAN-based approaches have shown impressive results for this task. In most applications, the network is trained based on a large number of paired images from the two domains. However, in some cases these correspondences are not known a priori, and no pairs of images are available for training. To keep using GANs, an alternative training procedure is necessary.

Under the adversarial training paradigm, we have two actors: the image-to-image network (generator), and a support network (discriminator). The discriminator tries to tell apart real images from those created by the generator, while the Generator is trained to deceive the discriminator [23]. The method proposed by Zhu et al. [18] creates automatically pairs of images that map into one another, making up for the lack of true images pairs. With reference to the architecture of Fig.2, input image $A$ is first transformed from its original domain $D_A$ to the output domain $D_B$ using generator $G_A$. Then, image $B$ is mapped back to the original domain $D_A$ by generator $G_B$. In this process, besides the usual adversarial losses associated with the discriminators, which ensure that generated images fit well the new domain, two additional *cycle consistency losses* are used, which ensure that $A = G_A(G_B(A))$, namely that the two images form a corresponding pair.

In Fig.1 we show a sample result of this method applied to satellite images. The image on the left is generated by the algorithm, while the other one is downloaded from Google Maps. At a first glance, no hints of a possible manipulation appear. Some more examples are reported in Fig.4, where original images and shown side by side with new versions obtained through a change of context: a different painting style (Van Gogh instead of Monet) or a different season (winter instead of summer). Modifying the context of an image is a very powerful editing tool, since major changes can be introduced without causing inconsistencies in illumination, shadows or perspective. Telling apart these fakes from original images can be very difficult for people with no specific training. Hence, this approach has clearly the potential for creating credible fake news (Fig.3).

## III. Detectors under Test

We compare a number of methods for the detection of image-to-image translation. Some are methods proposed for the detection of generic image manipulations [3], [24], [25] or specifically designed for computer graphics image detection [13]. Other are based on very deep state-of-the-art CNN architectures [26]–[28], pre-trained on ImageNet, which we fine-tuned to detect image-to-image translation. In addition, as a natural baseline, we consider the very same discriminator used by the authors of [18] in their GAN architecture. These methods are briefly described in the following.

**GAN discriminator.** As a baseline, we consider the discriminator used in [18] as part of the GAN. Its structure is shown in Fig.2 and has to be retrained on the available dataset to avoid biases.

**Steganalysis features.** Originally proposed for steganalysis [29], they have been successfully used for image forgery


Figure 3. Our applicative scenario. A synthetic image, generated through GAN-based image-to-image translation, is used to support fake news, spread out through Twitter. A properly trained classifier is needed to tell apart fake from original images.

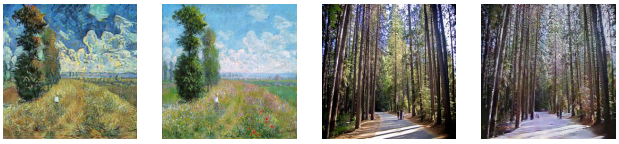

Figure 4. Examples of image-to-image translation [18]. Left: Van Gogh or Monet? Right: summer or winter?

detection [3]. They are extracted from high-pass residual images and measure the co-occurrences of detail micropatterns. Here, following [3], we use the best-performing single model, among the many proposed in [29], followed by a linear SVM classifier.

**Cozzolino2017.** In [24] an *ad hoc* CNN is designed to extract the very same features described above. Structural constraints are then removed, and the network is fine-tuned to optimize performance. The technique works on small patches, hence we implemented a majority voting procedure to make decisions on larger images.

**Bayar2016.** Following similar concepts, in [25] a constrained CNN is designed, where the first convolutional layer is forced to be a bank of high-pass filters. Also in this case a majority voting is implemented.

**Rahmouni2017.** In [13] the problem of telling apart natural from computer generated images is considered. The authors compare some CNN-based solutions to optimize the discriminative features. We implement the network that achieved the best performance (Stats-2L), comprising two convolutional layers of 32 and 64 filters, respectively. Four global statistics (mean, variance, maximum and minimum) are computed for each second-layer feature map, giving rise to a 256-d feature vector for each patch. A Multi-layer perceptron is used as classifier, and majority voting on disjoint patches provides the image-level decision.

**DenseNet.** It is a very deep CNN architecture [26], based on the residual network (ResNet) proposed in [30], which pursues effectively feature propagation and reuse within the network. Unlike in early CNN architectures, for each layer of DenseNet, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers. By creating shortcuts between input and output, feature propagation and reuse is encouraged, and the vanishing-gradient problem is much alleviated. Moreover, the number of parameters is much reduced ensuring faster training. Like the other architectures tested here, DenseNet is pre-trained on ImageNet, and fine-tuned on our image-to-image translation dataset.

**InceptionNet v3.** The core concept of InceptionNet is the use of multi-resolution representations of the input. In the first version of this network [31] also known as GoogleNet, the so-called Inception module was proposed, which performs convolutions with the input using $1\times1$, $3\times3$, and $5\times5$ filters in parallel, and concatenates the obtained features maps at the output. A later version [27] of the same net applied and extended the same concept, providing largely improved results.

**XceptionNet.** This network [28] brings to the extreme some ideas of InceptionNet, adopting fully separable filters. In each layer, the 3d filtering of the input feature maps is implemented as a 1d depthwise convolution followed by a 2d pointwise convolution. The theoretical performance loss induced by this strong structural constraints is largely compensated for by the fact that the number of parameters to be learned is vastly reduced, freeing resources to be spent more effectively for other ends.

## IV. Experimental Results

**Dataset.** In order to train and validate the detectors under comparison, we built a large dataset of samples of different categories from image-to-image translation [18] using the code available on-line (https://github.com/junyanz/CycleGAN). For each category, the dataset includes both the real and the fake images. So, for example, the *apple2orange* subset includes all the original images of apples and oranges used to train the GAN, and the corresponding fakes (oranges and apples, respectively) generated by the GAN itself once trained. In Fig.5 we show examples of all categories. These manipulations can be further grouped on the basis of their content. A first group concerns the translation of natural images, and includes apple2image, horse2zebra and winter2summer. A second group is related to the generation

Figure 5. Categories of image-to-image translation included in the dataset. In parentheses, the number of available images, half real, half fake. From left to right: apple2orange (4028), horse2zebra (4802), winter2summer (4386), cityscapes (5950), facades (800), map2sat (2192), Ukiyoe (3000), Van Gogh (3000), Cezanne (3000), Monet (5144).

Table I
SCENARIO 1: RESULTS ON THE ORIGINAL UNCOMPRESSED DATASET.

| Accuracy | ap2or | ho2zeb | win2sum | citysc. | facades | map2sat | Ukiyoe | Van Gogh | Cezanne | Monet | average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Steganalysis feat. | 98.93 | 98.44 | 66.23 | 100.00 | 97.38 | 88.09 | 97.93 | 99.73 | 99.83 | 98.52 | 94.40 |
| GAN discr. | 69.84 | 90.77 | 52.31 | 99.87 | 98.38 | 90.44 | 66.19 | 96.01 | 98.97 | 84.16 | 83.58 |
| Cozzolino2017 | 99.90 | 99.98 | 61.22 | 99.92 | 97.25 | 99.59 | 100.00 | 99.93 | 100.00 | 99.16 | 95.07 |
| Bayar2016 | 99.26 | 99.77 | 50.36 | 76.02 | 89.75 | 79.70 | 77.23 | 99.00 | 98.70 | 88.89 | 84.86 |
| Rahmouni2017 | 88.60 | 99.20 | 51.30 | 72.03 | 90.38 | 72.35 | 97.83 | 99.97 | 100.00 | 97.26 | 85.71 |
| DenseNet | 79.05 | 95.77 | 67.68 | 93.80 | 99.04 | 78.30 | 99.50 | 97.71 | 99.93 | 89.83 | 89.19 |
| InceptionNet v3 | 84.95 | 94.78 | 58.76 | 99.41 | 93.99 | 70.54 | 99.80 | 98.77 | 99.87 | 89.89 | 89.09 |
| XceptionNet | 95.91 | 99.16 | 76.74 | 100.00 | 98.56 | 76.79 | 100.00 | 99.93 | 100.00 | 95.10 | 94.49 |
| average | 89.55 | 97.23 | 60.58 | 92.63 | 95.59 | 81.97 | 92.31 | 98.88 | 99.66 | 92.85 | 89.55 |

of images from labeled maps of cityscapes, building facades and satellite images. In this case, only the real photos and the generated ones are included in the dataset, while the handmade label maps, both real and generated, are not considered. The last group comprises the generation of paintings from real photos and vice-versa. Overall, the dataset includes more than 36k 256×256 color images.

**Protocol.** To assess the performance in a real scenario where the manipulation is unknown a priori, a leave-one-manipulation-out (LOMO) procedure is adopted. Therefore, at each iteration, all images referring to a certain category are set aside for validation while the remaining ones are used for training. By so doing, we ensure that the classifier does not adapt to features of a specific class of translations, but learns patterns that are shared by all images generated by this procedure, thus generalizing across different manipulations.

**Scenario 1.** In Tab.I, for all considered techniques, we show the accuracy obtained for each type of manipulation, together with their average (last column) weighted by the number of samples. Among the considered techniques, the highest average accuracy is obtained by Cozzolino2017. This shallow network provides near-perfect classification for all manipulations except winter2summer, for an average of 95.07%. However, the handcrafted features [29] also provide very good results, as well as XceptionNet, among the deep networks. In the last row of Tab.I we also report the average classification accuracy for each manipulation. This allows us to spot the most challenging cases, winter2summer and map2sat. The former is an image-to-image translation that mostly introduces illumination changes and color swaps to simulate the snow or the vegetation in natural landscapes, without drastically changing the image structures. Meanwhile, the latter takes as input a labeled map and generates from scratch a satellite image very similar to those of Google Maps. These images appear very realistic, and rarely contain visual artifacts. On the opposite side, the manipulations most easily detected are horse2zebra, and some painting style transfer. In all these cases, the new structures introduced in the images bring with them visual artifacts, which are clearly visible in the first case, while in the paintings are partially masked by the nice visual appearance. In this case an automatic classifier easily spots the fake images, while this is not the case for a human observer.

**Scenario 2.** The good results of the former experiment should be taken with caution. In fact, it is very unlikely that the original fake image generated by a malevolent user is directly available. More likely, it will be posted on a social network, in order to reach as many people as possible and make the fake news go viral. As images are uploaded, they typically undergo automatic compression, which destroys the subtle patterns exploited by most classifiers. Therefore, robustness is a major issue when dealing with image forgery detection. In Tab.II we show results in the presence of compression for the very same detectors tested before, that is,

Table II
SCENARIO 2: RESULTS ON TWITTER-LIKE COMPRESSED IMAGES WITH TRAINING MISMATCH.

| Accuracy | ap2or | ho2zeb | win2sum | citysc. | facades | map2sat | Ukiyoe | Van Gogh | Cezanne | Monet | average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Steganalysis feat. | 50.20 | 50.12 | 50.00 | 50.00 | 50.00 | 50.00 | 50.00 | 50.20 | 50.00 | 49.98 | 50.05 |
| GAN discr. | 50.15 | 55.24 | 49.91 | 50.50 | 50.38 | 50.27 | 48.24 | 48.90 | 49.04 | 47.88 | 50.19 |
| Cozzolino2017 | 50.00 | 50.00 | 50.00 | 50.00 | 50.13 | 50.05 | 50.00 | 50.30 | 50.00 | 50.00 | 50.03 |
| Bayar2016 | 64.55 | 81.40 | 50.05 | 50.62 | 84.50 | 54.15 | 54.93 | 89.90 | 97.37 | 70.57 | 67.42 |
| Rahmouni2017 | 70.43 | 98.54 | 50.02 | 82.71 | 95.00 | 50.00 | 82.43 | 88.13 | 100.00 | 94.42 | 81.30 |
| DenseNet | 76.59 | 90.32 | 54.17 | 65.29 | 96.63 | 61.69 | 96.38 | 91.56 | 99.73 | 87.87 | 79.76 |
| InceptionNet v3 | 83.85 | 86.92 | 52.67 | 98.66 | 93.99 | 60.00 | 95.81 | 93.52 | 99.57 | 88.45 | 85.44 |
| XceptionNet | 90.87 | 96.09 | 52.56 | 97.93 | 98.20 | 51.34 | 99.34 | 97.27 | 99.70 | 86.29 | 87.17 |
| average | 67.08 | 76.08 | 51.17 | 68.21 | 77.35 | 53.44 | 72.14 | 76.22 | 80.68 | 71.93 | 68.92 |

Table III
SCENARIO 3: RESULTS ON TWITTER-LIKE COMPRESSED IMAGES.

| Accuracy | ap2or | ho2zeb | win2sum | citysc. | facades | map2sat | Ukiyoe | Van Gogh | Cezanne | Monet | average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Steganalysis feat. | 79.39 | 90.02 | 56.66 | 92.17 | 73.62 | 69.39 | 65.83 | 95.30 | 94.73 | 80.89 | 81.09 |
| GAN discr. | 63.29 | 91.08 | 51.90 | 53.14 | 88.75 | 79.35 | 76.56 | 80.32 | 96.41 | 81.83 | 73.33 |
| Cozzolino2017 | 79.57 | 89.82 | 53.74 | 86.81 | 62.88 | 89.64 | 67.67 | 98.80 | 99.93 | 87.33 | 82.62 |
| Bayar2016 | 54.64 | 95.34 | 50.27 | 54.00 | 90.63 | 52.69 | 58.90 | 74.27 | 99.77 | 78.60 | 69.17 |
| Rahmouni2017 | 84.96 | 98.35 | 54.30 | 57.60 | 91.88 | 54.93 | 96.83 | 99.63 | 99.77 | 89.72 | 80.97 |
| DenseNet | 78.27 | 93.44 | 66.94 | 97.83 | 98.19 | 80.45 | 97.54 | 98.53 | 99.57 | 83.95 | 88.51 |
| InceptionNet v3 | 78.60 | 95.23 | 64.54 | 96.09 | 90.14 | 63.84 | 99.53 | 96.31 | 100.00 | 86.21 | 87.37 |
| XceptionNet | 93.52 | 93.77 | 67.07 | 95.11 | 99.22 | 67.97 | 99.66 | 95.18 | 99.97 | 84.02 | 89.03 |
| average | 76.53 | 93.38 | 58.18 | 79.09 | 86.91 | 69.78 | 82.81 | 92.29 | 98.77 | 84.07 | 81.51 |

the classifiers are still trained on original samples but tested on compressed ones. The compression adopted is Twitter-like, which means that we implemented the same JPEG compression (in terms of quantization table, chrominance sub-sampling, quality factor) applied when an image is tweeted. This simple routine operation impairs dramatically the performance of most detectors under comparison, especially techniques based on handcrafted features or shallow neural networks. Deep networks exhibit a higher robustness, especially XceptionNet, with an accuracy of 87.17%, only 7% worse than the uncompressed case. This indicates that such networks do not rely only on textural micropatterns but also on other features, which survive compression.

**Scenario 3.** Of course, in the last experiment we have considered a worst case, with mismatch between training and test set. When the goal is to detect fakes on social media, it is reasonable to train the classifiers directly on compressed images. The results obtained in this case are reported in Tab.III. The steganalytic features and Cozzolino2017 now perform reasonably well, but more than 10% points below the uncompressed case, because some information useful to spot the fakes went lost during the lossy compression and cannot be recovered. XceptionNet keeps providing the best performance, reaching an accuracy of 89.03%.

## V. CONCLUSIONS

We have presented a study on the detection of images manipulated by GAN-based image-to-image translation. Several detectors perform very well on original images, but some of them show dramatic impairments on Twitter-like compressed images. Robustness is better preserved by deep networks, especially XceptionNet, which keeps working reasonably well even in the presence of training-test mismatching. Future research, besides extending the analysis to more manipulations and detectors, will study cross-method performance, possibly after transfer learning, w.r.t. other synthetic image generators. Moreover, we will test the performance in real world scenarios involving different social networks.

## VI. ACKNOWLEDGEMENT

This material is based on research sponsored by the Air Force Research Laboratory and the Defense Advanced Re search Projects Agency under agreement number FA8750-16-2-0204. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory and the Defense Advanced Research Projects Agency or the U.S. Government.

## REFERENCES

[1] S. Nightingale, K. Wade, and D. Watson, "Can people identify original and manipulated photos of real-world scenes?"

*Cognitive Research: Principles and Implications*, pp. 2–30, 2017.

[2] V. Schetinger, M. Oliveira, R. da Silva, and T. Carvalho, "Humans are easily fooled by digital images," *Computers & Graphics*, vol. 68, pp. 142–151, 2017.

[3] D. Cozzolino, D. Gragnaniello, and L.Verdoliva, "Image forgery detection through residual-based local descriptors and block-matching," in *IEEE Conference on Image Processing (ICIP)*, October 2014, pp. 5297–5301.

[4] S. Fan, T.-T. Ng, B. Koenig, J. Herberg, M. Jiang, Z. Shen, and Q. Zhao, "Image visual realism: From human perception to machine computation," *IEEE Transactions on Pattern Analysis and Machine Intelligence, in press*, 2017.

[5] S. Lyu and H. Farid, "How realistic is photorealistic?" *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 845 – 850, 2005.

[6] R. Wu, X. Li, and B. Yang, "Identifying computer generated graphics via histogram features," in *IEEE ICIP*, 2011, pp. 1933–1936.

[7] S. Dehnie, H. Sencar, and N. Memon, "Digital image forensics for identifying computer generated and digital camera images," in *IEEE ICIP*, 2006, pp. 2313–2316.

[8] A. Dirik, S. Bayram, H. Sencar, and N. Memon, "New features to identify computer generated images," in *IEEE ICIP*, Oct 2006, pp. IV–433–IV–436.

[9] A. Gallagher and T. Chen, "Image authentication by detecting traces of demosaicing," in *IEEE CVPR Workshops*, June 2008, pp. 1–8.

[10] J.-F. Lalonde and A. Efros, "Using color compatibility for assessing image realism," in *IEEE ICCV*, Oct 2007, pp. 1–8.

[11] R. Zhang, R.-D. Wang, and T.-T. Ng, "Distinguishing photographic images and photorealistic computer graphics using visual vocabulary on local image edges," in *International Workshop on Digital Forensics and Watermarking*, Oct 2011, pp. 292–305.

[12] D.-T. Dang-Nguyen, G. Boato, and F. D. Natale, "Discrimination between computer generated and natural human faces based on asymmetry information," in *Eusipco*, 2012, pp. 1234–1238.

[13] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizeny, "Distinguishing computer graphics from natural images using convolution neural networks," in *IEEE WIFS*, 2017, pp. 1–6.

[14] E. de Rezende, G. Ruppert, and T. Carvalho, "Detecting computer generated images with deep convolutional neural networks," in *SIBGRAPI Conference on Graphics, Patterns and Images*, 2017, pp. 71–78.

[15] O. Holmes, M. Banks, and H. Farid, "Assessing and improving the identification of computer generated portraits," *ACM Transactions on Applied Perception*, vol. 13, pp. 1–12, 2016.

[16] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face: Real-Time Face Capture and Reenactment of RGB Videos," in *IEEE CVPR*, 2016, pp. 2387–2395.

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE CVPR*, 2017.

[18] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE ICCV*, 2017.

[19] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *IEEE CVPR*, 2017.

[20] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *NIPS*, 2017.

[21] N. Haouchine, F. Roy, H. Courtecuisse, M. Nießner, and S. Cotin, "Calipso: Physics-based image and video editing through cad model proxies," *arXiv preprint arXiv:1708.03748*, 2017.

[22] M. Zampoglou, S. Papadopoulos, and Y. Kompatsiaris, "Large-scale evaluation of splicing localization algorithms for web images," *Multimedia Tools and Applications*, vol. 76, no. 4, pp. 4801–4834, february 2017.

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[24] D. Cozzolino, G. Poggi, and L. Verdoliva, "Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection," in *ACM Workshop on Information Hiding and Multimedia Security*, 2017, pp. 1–6.

[25] B. Bayar and M. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *ACM Workshop on Information Hiding and Multimedia Security*, 2016, pp. 5–10.

[26] G. Huang, Z. Liu, L. van der Maaten, and K. Weinberger, "Densely connected convolutional networks," in *IEEE CVPR*, 2017, pp. 4700–1708.

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE CVPR*, 2016, pp. 2818–2826.

[28] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *IEEE CVPR*, 2017, pp. 1800–1807.

[29] J. Fridrich and J. Kodovský, "Rich Models for Steganalysis of Digital Images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR*, 2016, pp. 770–778.

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE CVPR*, 2015.