Proceedings of the OAGM Workshop 2022 DOI: 10.3217/978-3-85125-954-4-03

Image Forgery Detection and Localization Using a Fully Convolutional Network

David Fischinger, David Schreiber and Martin Boyer
Austrian Institute of Technology - AIT
{forename}.{surename}@ait.ac.at

Abstract

To fight the growing problem of fake news – and specifically image manipulation – we propose a simple, yet efficient neural network architecture for detecting and localizing various image forgeries on a pixel-level. Robust features for forgery detection and localization were learned and the trained model performs well, even on heavily downscaled images, but without the excessive processing time of competitive approaches based on image decomposition and merging of the fragmental results. We provide detailed explanations regarding the creation of our training dataset comprising 1.9 million images. Finally, we compare the proposed solution against several state-of-the-art methods on four public benchmark datasets in order to demonstrate its superior performance.

1. Introduction

”Fake News” are a growing problem of our society. Technological progress makes it easier and faster to produce high quality forgeries of digital media material such as audio, video and images. The impact ranges from satirical memes to orchestrated political Fake News campaigns aiming to influence public opinion – and at the same time raising the hard question where to draw a line between fighting Fake News and the fundamental right of free speech. In this paper, we present a new approach for identifying forged regions in images, thereby enabling institutions such as media organizations and interested citizens to get a better indication of whether specific images may have been manipulated.

During the last decade, various approaches for detecting the main categories of image forgery were proposed: copy-move [9] splicing [11], inpainting [8] and further specific filtering, subsumed as enhancement [20]. However, these approaches frequently focus on specific features of the respective manipulation type. In recent years, more general approaches for multiple manipulation types were developed, such as [24] and [23]. Each of them promotes sophisticated and problem-specific concepts, like modeling known and unknown noise on images that result from transmis-

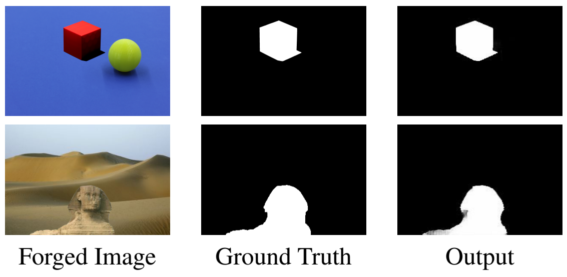


Table 1. Results of our model for image forgery detection and localization. Example images are taken from the `CASIA` [4] and the `NIST` [15] datasets.

sion to Online Social Networks (OSNs). In this paper, we present an image forgery detector which outperforms state-of-the-art approaches with a quite simple and general deep learning network architecture and a carefully constructed training dataset. To be more specific, our major contributions are as follows:

- We propose a deep learning network architecture for the task of image forgery detection and localization, capable to learn relevant features for composed image manipulations.
- We present a model that outperforms current state-of-the-art (SOTA) approaches on four public benchmark datasets.
- We present a processing time comparison with a SOTA approach showing a significant time saving, especially for larger images.
- We give a detailed description of our training dataset, as well as instructions on how to generate such a dataset.

2. Related Work

Many methods of detecting and localizing image forgery have been published (see, for example, the review of [21] and references therein), in order to ensure visual information authenticity. Some of these forensic techniques are designed to detect specific forms of tampering, such as splicing [11], copy-move [12, 16, 22, 25–27], and inpainting [8]. Unfortunately, these forensic approaches can only be applied to detect specific tampering manipulations.

In recent years, deep learning-based methods were developed to address the problem of detecting general (compound) types of forgeries. In [28], a two-stream Faster R-CNN network is trained end-to-end to detect the tampered regions in a manipulated image. One of the two streams is an RGB stream whose purpose is to extract features from the RGB image input. The other one is a noise stream that leverages the noise features extracted in order to discover noise inconsistencies between authentic and tampered regions. Notably, [24] proposes a unified deep neural architecture called ManTra-Net, which is an end-to-end network that performs both detection and localization without extra preprocessing and postprocessing. ManTra-Net is a fully convolutional network which can handle images of arbitrary sizes and many known – and even unkown – forgery types. Furthermore, the authors design a self-supervised learning task to learn robust image manipulation features, formulate the forgery localization problem as a local anomaly detection problem, and propose a long short-term memory (LSTM) solution to assess local anomalies.

In [13], a CNN-based image forgery detection framework is proposed which makes decisions based on full-resolution information gathered from the entire image, without the need for preliminary image resizing. The framework is trainable end-to-end with limited memory resources and weak (image-level) supervision, thus allowing for the joint optimization of all parameters. The work of [29] addresses the issue of tampering localization by focusing on the detection of commonly used editing tools and operations in Photoshop. A fully convolutional encoder-decoder architecture is designed, as well as a training data generation strategy by resorting to Photoshop scripting.

The widespread availability of online social networks (OSNs), e.g., Twitter, Facebook, Whatsapp, etc., makes them the dominant channels for transmitting forged images. However, almost all OSNs manipulate the uploaded images in a lossy fashion (including format conversion, resizing, enhancement filtering and JPEG compression). The noise introduced by these lossy operations could severely affect the effectiveness of forensic methods. In a recent paper [23], the problem of OSN-shared image forgeries is tackled by employing a dedicated training scheme. A baseline detector is presented, which is based on a modified U-Net [17] as the backbone architecture. Next, an analysis of the noise introduced by OSNs is conducted, and the noise is decoupled into two parts, i.e., predictable noise and unseen noise. These are then modelled separately and the modelled noise is further incorporated into the training framework.

Outline: The rest of this paper is structured as follows: Section 3 describes in detail how the datasets for training and validation were generated. In section 4, we present different models we have created, evaluate them on benchmark datasets, and describe the architecture of the best performing model in detail. In section 5, our proposed network is evaluated and compared to state-of-the-art methods. Final remarks are made in section 6.

3. Datasets

Currently, there are no sufficiently large training datasets publicly available for the task of image forgery detection. In the following, a detailed description is provided, of how our training dataset, which comprises 1.9 million manipulated images, was created.

3.1. Training and Validation Datasets

As a source of pristine and donor images we facilitated the MS-Coco [10] 2017 training dataset containing 118K images. This public and widely used dataset encompasses a wide range of images. Our training dataset includes 4 major types of image manipulation: splicing, copy-move, removal and enhancement. The overall process for training data generation was as follows:

1. Select Pristine Image:
   A pristine image $\mathcal{I}_P$ from MS-COCO 2017 was selected randomly. For the few images with width W or height H smaller than 224 pixels, the image was resized to the size $(max(W, 224), max(H, 224))$. For 50% of the images $\mathcal{I}_P$ in the training dataset, a proportion-preserving downscaling was executed. This avoided extracting only small portions of bigger images (like a monochrome patch depicting a part of the sky from the original image). This scaling for an image $\mathcal{I}_P$ with size (W, H) to (W_{new}, H_{new}) was done as follows:

$$\begin{aligned} W_{new} &= max(\lfloor(\frac{224 \cdot W}{min(W,H)})\rceil, 224) \\ H_{new} &= max(\lfloor(\frac{224 \cdot H}{min(W,H)})\rceil, 224) \\ \mathcal{I}_P &= \mathcal{I}_P.resize((W_{new}, H_{new})) \end{aligned} \tag{1}$$

   Next, a patch of size $(224, 224)$ pixels is randomly chosen from the image $\mathcal{I}_P$ and used as a pristine image patch $\mathcal{P}$.

2. Select Donor Image:
   A donor image $\mathcal{I}_D$ from MS-COCO was selected. For the splicing operation, a random image other than the pristine image $\mathcal{I}_P$ was selected. For the copy-move, removal and enhancement manipulations, the same pristine image was selected as a donor image

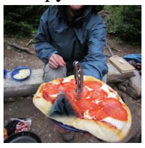
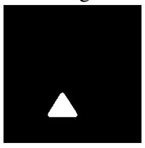
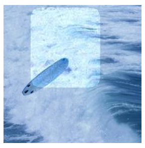
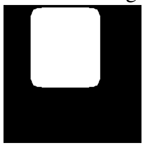
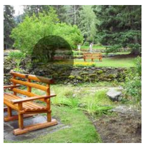
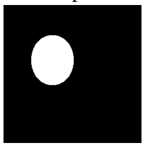
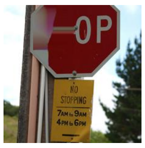
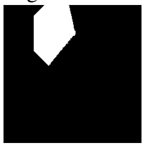
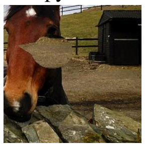
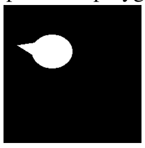
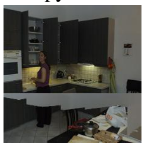
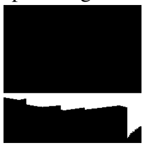
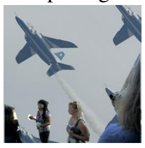
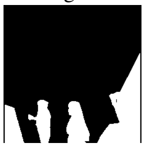
Forgery Type	**Manipulation Mask Shape**
Copy-Move 	triangle 
Enhance 	rounded rectangle 
Enhance 	ellipse 
Removal 	poligon 5 vertices 
Copy-Move 	ellipse + 4V polygon 
Copy-Move 	superpixel segmentation 
Splicing 	person segmentation 

Table 2. Training and Validation data: showing forged images for major manipulation types and related manipulation mask shapes

($\mathcal{I}_D = \mathcal{I}_P$). Then, a donor patch $\mathcal{D}$ of size $(224, 224)$ was randomly cropped from $\mathcal{I}_D$. For enhancement and removal (inpainting) manipulations, the donor patch $\mathcal{D}$ and the pristine patch $\mathcal{P}$ share the same location in $\mathcal{I}_D = \mathcal{I}_P$.

3. Preprocess Donor Image Patch $\mathcal{D}$:
Table 4 shows which preprocessing steps may be applied to the donor image patch $\mathcal{D}$ for each manipulation type. **Resample** rescales the height and the width image dimension independently by 70 to 130 percent. The resulting image has at least the size $(224, 224)$. The preprocessing step **Flip** flips the donor image horizontally with a likelihood of 50%, while **Rotate** rotates the image by either 90, 180 or 270 degrees with a likelihood factor controlled by a parameter (for the generated dataset, 30% of the donor images were rotated). **Blur** is blurring the donor image with a likelihood of 50%. In case the blurring filter is applied, either `ImageFilter.BoxBlur` or `ImageFilter.GaussianBlur` from the Python package `PIL` are used with equal probabilities. The blur radius is set randomly between 1 and 7 pixels. **Contrast** uses one of the ImageFilters `EDGE_ENHANCE`, `EDGE_ENHANCE_MORE`, `SHARPEN`, `UnsharpMask` or `ImageEnhance.Contrast` from the Python package PIL. **Noise** adds Gaussian noise with mean and standard deviation $(\mu, \sigma) = (0, 12)$ with likelihood of 1 out of 3. The **Brightness** is changed with probability of 50% by a factor uniformly chosen from the range [0.5-1.5]. With 0.5 probability, a **JPEG-Compression** with quality factor $10x$ for $x \in [1, 2, 3, 4, 5, 6, 7]$ is employed. In case the manipulation type is **Removal**, an inpainting filter from OpenCV [2] is applied (either `cv2.INPAINT_TELEA` or `cv2.INPAINT_NS`) on the manipulation mask defined in the next step.
In case the chosen manipulation type is **Enhance** and none of the filters (blur, contrast, noise, brightness, jpeg compression) were applied to the donor patch $\mathcal{D}$, the process is repeated.

4. Create Binary Manipulation Mask
7 types of binary masks were used to define the region in an image where manipulations have been executed (see Tab. 3). In Table 2, various examples for created masks and the resulting forged images are shown. The Python's image processing toolbox scikit-image is employed to segment the donor patch in Superpixels [1] of appropriate size, and selects one Superpixel (connected set of pixels) for the splicing manipulation. The "person segmentation" uses the segmentation ground truth from the MS-COCO dataset. All pixels from a donor image patch $\mathcal{D}$ marked as person are selected and used as splicing input. Masks are recalculated if their portion of the image patch is not in the range of 5% to 40%.

5. Generate Forged Image
Given a pristine patch $\mathcal{P}$, a donor patch $\mathcal{D}$, a manipulation m and a binary manipulation mask $\mathcal{M}$, the forged

Shape of Mask	Parameters	Impact
Triangle	p1, p2, p3	3 random points
Rounded Rectangle	X,Y, r	2 points for Bbox; radius of the corners
Ellipse	X, Y	2 points to define the bounding box
Polygon with 5 vertices	p1,. . . ,p5	sequence of 5 random points
Ellipse + Polygon with 4 vertices	X,Y, p1,..,p4	ellipse + 4 vertex polygon
Superpixel Segmentation	[min, max]	range for number of Superpixels
Person Segmentation	-	

Table 3. Types of mask shapes generated for local image manipulation

image $\mathcal{X}$ is given by

$$\mathcal{X} = \mathcal{M} \cdot \mathcal{P} + (1 - \mathcal{M}) \cdot m(\mathcal{D}) \qquad (2)$$

meaning that each pixel of the resulting image $\mathcal{X}$ is taken either from the pristine patch $\mathcal{P}$ or the manipulated donor patch $\mathcal{D}$, depending on the binary mask $\mathcal{M}$. In case of a copy-move manipulation, an additional translation of the copied image part $(1 - \mathcal{M}) \cdot m(\mathcal{D})$ towards another position in the pristine image patch is made.

Using this process, a training dataset with 1.9 million forged images was generated, comprised of $700,000$ splicing, $500,000$ copy-move, $400,000$ enhance and $200,000$ inpainting images as their main forgery type. This training dataset was used in Section 4 for model training.

Manipulation-Type	C	S	R	E
Resample	×	×	–	–
Flip	×	×	–	–
Rotate	×	×	–	–
Blur	–	–	–	×
Contrast	–	–	–	×
Noise	–	–	–	×
Brightness	–	–	–	×
JPEG-Compression	–	–	–	×

Table 4. Preprocessing steps for donor image per manipulation types: Copy-Move (**C**), Splicing (**S**), Removal (**R**) and Enhancement (**E**)

4. Network Architecture Evaluation

In this section we implemented several network architectures for image forgery detection and localization. The models were trained on the dataset created in Sec. 3. The problem of image forgery detection and localization is essentially a segmentation problem in which each pixel is classified as an original or a manipulated pixel. For this task, U-Nets are a well established network architecture and we present 3 variants of U-Net models with promising performance, evaluate them on 4 benchmark datasets and describe the best performing model in more detail.

4.1. Models and Evaluation

MobileNet - MoNet: This model is a modified U-Net. U-Nets consist of an encoder for downsampling and a decoder for upsampling. MobileNetV2 [19], pretrained on Imagenet, is used as an encoder. MobileNet [5] is a lightweight architecture that has already learned robust features in the context of image classification and hence allows to reduce the number of trainable parameters. For upscaling, the Tensorflow implementation of pix2pix [7] was utilized. Furthermore, 5 skip connections between output layers from downsampling and layers form the upsampling part were established.
U-NET: This network is one implementation of the original U-Net architecture [17].
SE-UN: Our improved version of **U-NET** architecture, which adds a recalibration with Spatial and Channel Squeeze & Excitation Blocks [18].

Table 5 shows results for the three network architectures evaluated on the benchmark datasets `CASIA` [4], `Columbia` [6], `DSO` [3] and `NIST16` [15].

While the MobileNet implementation (MoNet) gives the best results for the metrics F1 and IoU averaged over all 4 benchmark datasets, SE-UN performs better for AUC and the pixel-wise accuracy. Since the average over all 4 metrics is higher for the latter model (0.574) compared to MoNet with a score of 0.566, we chose our U-Net variation with additional Spatial Channel Squeeze and Excitation (SE-UN) for further experiments and SOTA comparison. The architecture is depicted in more detail in Fig. 1.

4.2. Implementation Details

The deep learning framework Tensorflow was used for training our network. For training and detection, the images were resized to $(224, 224)$ pixels. An `Nvidia GeForce GTX 1080` Ti GPU was used for training, with batch size set to 16. We use Adam optimizer and perform 1500 steps per epoch and stop after the loss of the validation dataset did not improve for 35 epochs. Training starts with a learning rate of 0.00006, which is halved after 20 epochs without improvement.

Models	Test Datasets																				
	DSO [3]				Columbia [6]				NIST [15]				CASIA [4]				Average				
	AUC	F1	IoU	ACC	AUC	F1	IoU	ACC	AUC	F1	IoU	ACC	AUC	F1	IoU	ACC	AUC	F1	IoU	ACC	all
MoNet	.690	**.348**	**.227**	.716	.781	**.663**	**.568**	**.829**	.660	.257	.195	.833	.723	.384	.306	.878	.713	**.413**	**.324**	.814	.566
U-NET	.599	.098	.061	.835	.803	.519	.411	.802	.655	.222	.174	.897	.750	.212	.176	.924	.701	.263	.206	.864	.508
SE-UN	**.732**	.152	.108	**.848**	**.827**	.503	.428	.827	**.780**	**.265**	**.221**	**.921**	**.851**	**.429**	**.369**	**.929**	**.797**	.337	.282	**.881**	**.574**

Table 5. Comparison of three developed U-Net architectures (MoNet, U-NET, SE-UN) by AUC, F1 and IoU metrics.

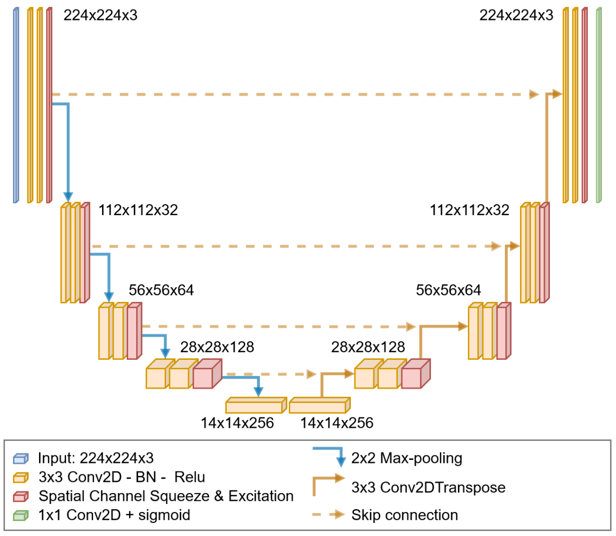


Figure 1. Our proposed network: A U-Net architecture with 4 skip connections and spatial channel Squeeze & Excitation (scSE) extension. Two (3x3)-convolutions combined with one scSE layer, a batch normalization (BN) layer and a Relu activation layer form the building blocks, followed by Max-pooling (encoder) respectively upscaling (with a Conv2DTranspose layer in the decoder part on the right side). The expected input image size $(H, W) = (224, 224)$.

4.3. Image Manipulation Classification

To investigate the capability of our networks to handle the image manipulation classification (IMC) task, we trained the encoder part of our MoNet model with an additional Softmax layer to detect one of the 4 main manipulation types (splicing, copy-move, removal, enhancement) as the outcome. We trained on a dataset created according to Sec. 3 with one million images divided into 4 classes. For an evaluation dataset with 1,200 images created similarly to the training dataset, a classification accuracy of $94,92\%$ was achieved, thus showing the capacity of the model for the classification task.

5. Experimental Evaluation

5.1. SOTA Comparison

The proposed model **SE-UN** was compared with 4 state-of-the-art methods: ForSim [14], DFCN [29], ManTra-Net [24] and OSN [23]. We used the officially released models from the latter two approaches to evaluate the methods on the four benchmark datasets CASIA V1 [4], Columbia [6], DSO [3] and NIST16 [15]. For DFCN and ForSim, we listed the results from [23]. As metric, the Area Under the Receiver Operating Characteristic curve (AUC) was chosen as it is widely used in the research field of image forgery detection. As in previous works (e.g. [23]), the ground truth mask is inverted if it sums up to more than 50%) of the image. This seems in line with the principal concept of manipulation detection, although it has an insignificant impact on the overall metric scores.

As shown in Table 6, our approach performed best on the Columbia, NIST16 and CASIA datasets. Only for the DSO dataset, the ForSim achieved the highest AUC value. With an average AUC-value of 79.7 our approach outperformed OSN, the second best performing approach, by 5.3 points. Table 7 shows examples from each of the benchmark datasets, comparing the three methods with the highest average AUC values.

5.2. Processing Time

Our proposed SE-UN model is trained on images of size $(224, 224)$. Therefore, for the purpose of evaluation, images are first rescaled to this size. The learned network features are so robust, that they are capable to predict forgeries with SOTA performance even on down-scaled images. This brings significant advantages compared to other approaches ([13], [23]), which make decisions base on full resolution information gathered from whole images. In Table 8, we show a comparison with [23] of the processing time when predicting all images for each of the benchmark datasets. For datasets with images of smaller size (CASIA, Columbia), the processing time of our approach and the OSN [23] method is on the same scale. For datasets with larger images (DSO, and specifically NIST), the processing time for OSN rises rapidly with the size of the images. The reason is that, for the 564 images of the NIST dataset, this approach produces 24,996 tiles from the original images, executes forgery detection on each of these image parts, and finally merges the result for the predicted outcome per image.

Models	AUC of Test Datasets				
	DSO [3]	Columbia [6]	NIST [15]	CASIA [4]	Average
ForSim [14]	**.796**	.731	.642	.554	.681
DFCN [29]	.724	.789	.778	.654	.736
ManTra-Net [24]	.795	.747	.634	.776	.738
OSN [23]	.723	.815	.686	.751	.744
SE_UN (ours)	.732	**.827**	**.780**	**.851**	**.797**

Table 6. Comparison of our **SE_UN** model with SOTA methods using AUC metric on four benchmark datasets. The highest value per column is **bold**

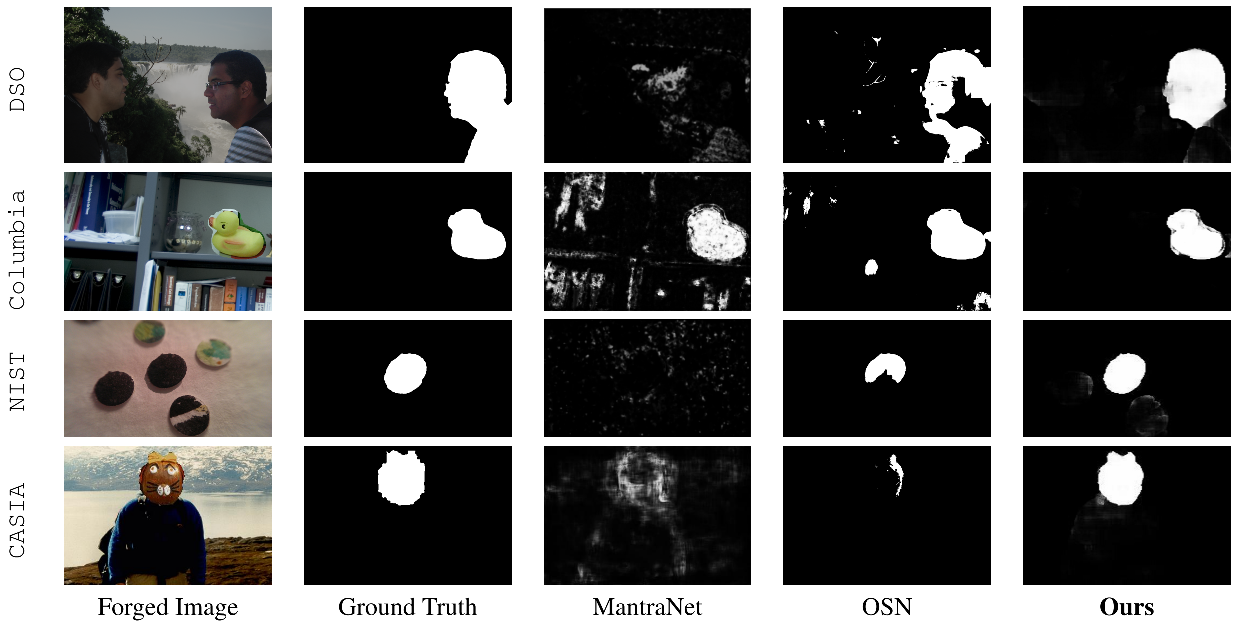


Table 7. Examples of qualitative comparison of MantraNet [24], OSN [23] and our proposed forgery detector. Each line shows one example image for each of the four benchmark datasets DSO [3], Columbia [6],NIST [15], CASIA [4]. The five columns show: the forged image (input), manipulated area (ground truth), results (output) from MantraNet, OSN and our detector.

Dataset	# Images	Format	t-OSN	t-Ours
CASIA [4]	920	jpg	169	**94**
Columbia [6]	160	tif	**120**	178
DSO [3]	100	png	701	**20**
NIST [15]	564	jpg	15250	**188**

Table 8. Processing time (t) in seconds for prediction per benchmark dataset. For datasets with huge images as NIST (images of size up-to 5616×3744 pixels) tile-based approaches considerably take longer than approaches performing pre-scaling.

6. Conclusion

In this paper, we propose a new network model for image forgery detection. The proposed approach reaches and exceeds state-of-the-art performance on various benchmark dataset. The relatively simple network architecture learns very robust features from scratch from the presented dataset. Even on heavily down-scaled images, the detector delivers very good results, and a considerable processing time advantage for bigger sized images compared to competitors.

Our model can detect compound and unseen forgeries of postprocessed images (as included in the benchmark datasets). But still, the fact that our model achieves pixel-wise accuracy rates of 99% on a validation dataset created similarly to the training dataset, but about 88% on the benchmark datasets used for evaluation shows potential for more improvement of the detector by generating more challenging training data.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpix-

els compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. 3

[2] Gary Bradski. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000. 3

[3] T. Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. Rezende Rocha. Exposing digital image forgeries by illumination color classification. *IEEE Trans. Inf. Forensics and Security*, 8(7):1182–1194, 2013. 4, 5, 6

[4] J. Dong, W. Wang, and T. Tan. Casia image tampering detection evaluation database. In *IEEE China Summit Inter. Conf. Signal Info. Proc.*, pages 422–426. IEEE, 2013. 1, 4, 5, 6

[5] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, 2017. 4

[6] Y. Hsu and S. Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *IEEE Inter. Conf. Multim. Expo*, pages 549–552. IEEE, 2006. 4, 5, 6

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 4

[8] Haodong Li, Weiqi Luo, and Jiwu Huang. Localization of diffusion-based inpainting in digital images. *IEEE Transactions on Information Forensics and Security*, 12(12):3050–3064, 2017. 1, 2

[9] L. Li, S. Li, Hancheng Zhu, Shu-Chuan Chu, John Roddick, and Jeng-Shyang Pan. An efficient scheme for detecting copy-move forged images by local binary patterns. *Journal of Information Hiding and Multimedia Signal Processing*, 4:46–56, 01 2013. 1

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. Microsoft coco: Common objects in context. In *ECCV*, September 2014. 2

[11] Siwei Lyu, Xunyu Pan, and Xing Zhang. Exposing region splicing forgeries with blind local noise estimation. *International Journal of Computer Vision*, 110:202–221, 11 2013. 1, 2

[12] Toqeer Mahmood, Aun Irtaza, Zahid Mehmood, and Muhammad Mahmood. Copy-move forgery detection through stationary wavelets and local binary pattern variance for forensic analysis in digital images. *Forensic Science International, Elsevier*, 279:8–21, 10 2017. 2

[13] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. A full-image full-resolution end-to-end-trainable cnn framework for image forgery detection. *IEEE Access*, PP:1–1, 07 2020. 2, 5

[14] Owen Mayer and Matthew C Stamm. Forensic similarity for digital images. *IEEE Transactions on Information Forensics and Security*, 2019. 5, 6

[15] National Institute of Standards and Technology (NIST). Nist nimble 2016 datasets. `https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation/`. 1, 4, 5, 6

[16] Junlin Ouyang, Yizhi Liu, and Miao Liao. Robust copy-move forgery detection method using pyramid model and zernike moments. *Multimedia Tools and Applications*, 78:1–19, 04 2019. 2

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 2015. 2, 4

[18] Abhijit Guha Roy, Nassir Navab, and Christian Wachinger. Recalibrating fully convolutional networks with spatial and channel 'squeeze & excitation' blocks. 4

[19] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 4

[20] Jee-Young Sun, Seung-Wook Kim, Sang-Won Lee, and Sung-Jea Ko. A novel contrast enhancement forensics based on convolutional neural networks. *Signal Processing: Image Communication*, 63:149–160, 2018. 1

[21] Luisa Verdoliva. Media forensics and deepfakes: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):910–932, 2020. 1

[22] Yuan Wang, Lihua Tian, and Chen Li. Lbp-svd based copy move forgery detection algorithm. In *2017 IEEE International Symposium on Multimedia (ISM)*, pages 553–556, 2017. 2

[23] Haiwei Wu, Jiantao Zhou, Jinyu Tian, Jun Liu, and Yu Qiao. Robust image forgery detection against transmission over online social networks. *IEEE Transactions on Information Forensics and Security*, 2022. 1, 2, 5, 6

[24] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9535–9544, 2019. 1, 2, 5, 6

[25] Pei Yang, Gaobo Yang, and Dengyong Zhang. Rotation invariant local binary pattern for blind detection of copy-move forgery with affine transform. In *Cloud Computing and Security*, pages 404–416, 07 2016. 2

[26] Ibrahim A. Zedan, Mona M. Soliman, Khaled M. Elsayed, and Hoda M. Onsi. Copy move forgery detection techniques: A comprehensive survey of challenges and future directions. *International Journal of Advanced Computer Science and Applications*, 12(7), 2021. 2

[27] Jun-Liu Zhong and Chi-Man Pun. An end-to-end dense-inceptionnet for image copy-move forgery detection. *IEEE Transactions on Information Forensics and Security*, 15:2134–2146, 2020. 2

[28] Peng Zhou, Xintong Han, Vlad Morariu, and Larry Davis. Learning rich features for image manipulation detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2018. 2

[29] Peiyu Zhuang, Haodong Li, Shunquan Tan, Bin Li, and Jiwu Huang. Image tampering localization using a dense fully convolutional network. *IEEE Transactions on Information Forensics and Security*, 16:2986–2999, 2021. 2, 5, 6