Augment and Adapt: A Simple Approach to Image Tampering Detection

Yashas Annadani C.V. Jawahar
Centre for Visual Information Technology (CVIT)
IIIT Hyderabad, India

***Abstract*—Convolutional Neural Networks have been shown to be promising for image tampering detection in the recent years. However, the number of tampered images available to train a network is still small. This is mainly due to the cumbersomeness involved in creating lots of tampered images. As a result, the potential offered by these networks is not completely exploited. In this work, we propose a simple method to address this problem by augmenting data using inpainting and compositing schemes. We consider different forms of inpainting like simple inpainting and semantic inpainting as well as compositing schemes like feathering in order to augment the data. A domain adaptation technique is employed to reduce the domain shift between the augmented data and the data available using proprietary softwares. We demonstrate that this method of augmentation is effective in improving the detection accuracies. We present experimental evaluation on two popular datasets for image tampering detection to demonstrate the effectiveness of the proposed approach.**

I. Introduction

Due to the ever growing number of powerful softwares and algorithms available for the manipulation of images, it has become relatively easy to tamper an image for unethical and illegal purposes. As a result, research in image tampering detection has been receiving considerable interest [1][2][3][4]. Recently, Convolutional Neural Networks (CNN) have emerged as a popular choice for image tampering detection due to their success in image recognition tasks like object classification [5][6] as well as their ability to directly learn from data without involving any complicated hand-crafting [3][2][7]. However, the advantages offered by CNNs are not completely exploited for detection of tampering in images. The major setback for efficient usage of CNN is the fact that the amount of training data available in terms of number of tampered images is still very small. The biggest dataset for image tampering detection is the CASIA TIDE V2 [8] dataset with around 5,000 tampered images. However, this is still small for a large CNN with potentially millions of learnable parameters. In order to circumvent this problem, existing methods either resort to smaller architectures [2] or employ networks whose layers are derived from existing filter based approaches [7][9]. Most of these methods use patches derived from full images in order to augment the dataset. While this is a plausible approach, the amount of data available even after this augmentation process might not be sufficient. In addition, sampling patches from overlapping regions can lead to less variance in the input data, and thus the resulting model might end up picking cues from color and texture which might not be the most relevant.

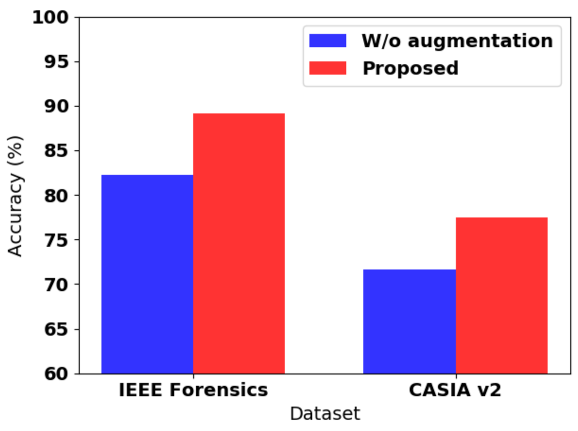

Fig. 1: Significant quantitative gain is observed using the proposed approach on two of the standard image tampering detection datasets - IEEE Forensics [10] and CASIA TIDE V2 [8].

In this work, we propose a simple yet effective data augmentation method specific to image tampering detection in order to alleviate the aforementioned problems. We posit that an inpainted image or a composited image is essentially a tampered image. This can be validated by a simple fact that they share the same underlying principles, albeit with a different motivation. For example, *object removal* form of tampering essentially involves replacing the pixels of an object of interest with pixels which blend with the background. Inpainting adopts the same principle. This brings to forefront the following question: Is it possible to leverage them for detecting tampered images? The answer to this question is not obvious and has not been explored yet in the literature. We demonstrate that through a systematic process of generating the inpainted and composited images, the answer is an emphatic yes. This is the primordial contribution of this work. Although it is to be expected that the tamperings generated by inpainting and compositing might be slightly different from the ones produced using proprietary softwares, employing a simple domain adaptation method of minimizing the Maximum Mean Discrepancy (MMD) [12] between these two forms of data can help to learn an efficient representation of tamperings. Our contributions are:

1) We propose inpainting and compositing as data augmentation strategies for image tampering detection in a CNN

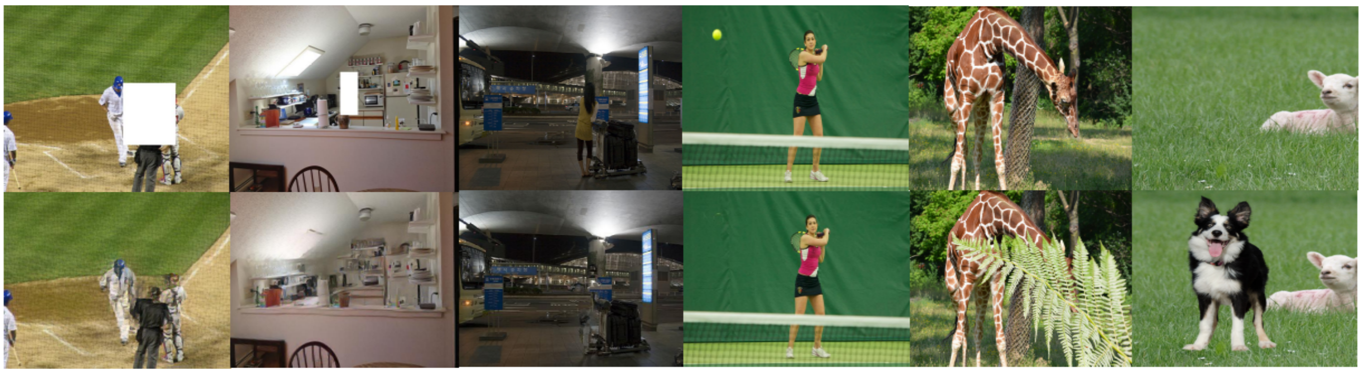
Fig. 2: Examples from each of the tampering schemes involved in augmenting the data. The first row corresponds to the original images and the second row corresponds to inpainted or composited images. The first two columns correspond to augmentation using simple inpainting, the next two columns correspond to augmentation using semantic inpainting and the last two columns correspond to feathering. The images are taken from COCO [11] dataset.

framework which helps improve the image tampering detection.

2) The proposed approach is an end-to-end CNN framework which does not involve any hand-crafted filters.
3) Extensive experimental evaluation on two benchmark datasets demonstrates that the proposed data augmentation method when combined with a state-of-the-art domain adaptation method surpasses the state-of-the-art for image tampering detection.

The rest of the paper is organized as follows. In Section II, we present a brief overview of the related work. Section III describes the proposed approach in detail. Section IV describes the experimental evaluation and results followed by Section V, which provides the conclusion.

II. Related Work

Deep learning, especially in the form of Convolutional Neural Networks has been used in the literature for image tampering detection in various forms. Chen et al. [7] employ CNN for median filtering based image forensics. The first layer is a fixed layer which computes the median residuals of the input image. Bayar et al. [9] proposed a new convolutional layer consisting of filters which produce a prediction error of the center pixel with respect to its neighborhood. In [13], the weights of the first layer are initialized with the weights of spatial rich model [14] high pass filter residuals. The experiments demonstrate that this method of initialization is superior to the standard deep learning initialization methods like Xavier [15]. Although the results of these methods are encouraging, they still rely on hand-crafted filters for learning representations of tamperings. Our network architecture is similar to these approaches. However, our approach does not need any hand-crafted filters and is entirely end-to-end.

Bondi et al. [16] employ CNN to extract camera model features of different patches of an image. A suitable clustering algorithm is used to aggregate patches which have similar camera model features. An image is labeled as pristine if all of its patches belong to the same cluster, else it is classified as tampered. Rota et al. [2] employ a CNN network which consists of two convolutional layers followed by two fully connected layers. The authors extend their detection framework to localization by employing an edge detector to mine candidate tampered regions. Recently, Bappy et al. [3] employ a combination of convolutional and Long-Short Term Memory (LSTM) layers to detect as well as segment the tampered regions. The above methods rely on patch based sampling for training, as the number of tampered images available is less. In our approach, we augment the data using inpainting and compositing to address the problem.

Our work is focused on effective data augmentation strategies for data-hungry convolutional neural networks. Although this has not been explored in the context of image tampering detection, there exist multiple works which have attempted to address the data scarcity problem for object classification and other recognition tasks. Chen et al. [17] augment the data for describing people based on their clothing in unconstrained scenario by obtaining images from online shopping sites which are taken in a constrained environment. The authors employ a two-stream CNN network for the task. A domain adaptation method is developed to bridge the domain disrepancy between the constrained online shopping data and the unconstrained images containing people. Our approach follows similar lines, although in the context of image tampering detection. However, unlike online shopping sites, tampered image data is relatively tougher to augment as it does not involve any semantics nor any other observable cues thereby. Dwibedi et al. [18] employ splicing operation of object instances onto random backgrounds to synthesize training examples for instance detection. Similar synthesis and augmentation approaches coupled with adaptation has been proposed for various tasks like viewpoint refinement [19] and semantic segmentation [20]. However, this paradigm of augmenting and then performing adaptation is not explored for the task of image tampering detection.

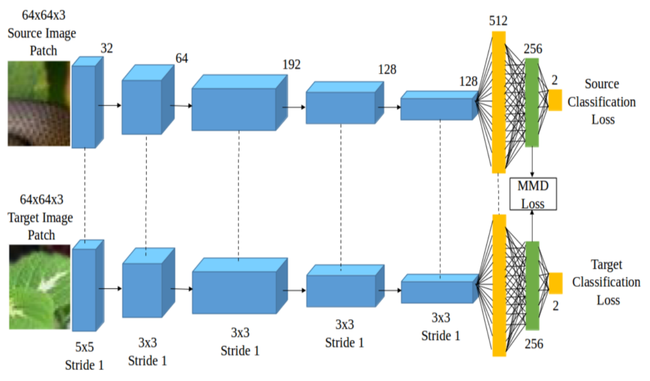

Fig. 3: An overview of the network used for learning artefacts of tampering. The connection between layers using a dotted line indicates that the corresponding layers share parameters. The final three layers are fully connected layers.

III. Proposed Method

Availability of large datasets like ImageNet [21] was one of the major factors for the success of convolutional neural networks for recognition tasks in computer vision [5]. Though image tampering detection is essentially a two class classification task, the artefacts are subtle and hence are difficult to model. As a result, large amount of data especially in terms of number of tampered images is required to train a deeper network. However, manually obtaining a lot of tampered images is cumbersome. Therefore we seek alternate forms of generating tampered images which are very similar to the ones produced using proprietary softwares and which contain similar artefacts, thus helping in augmenting the data.

A. Augmenting Data using Inpainting and Compositing

In this work, we focus on the most common forms of tampering - *instance removal*, *copy and paste* and *splicing*. These forms of tampering can be attributed to two different operations: 1) Inpainting [22] for *instance removal* and 2) Compositing [23] for *copy and paste* as well as *splicing*. With this insight, we systematically generate inpainted images and composites which can be easily employed to create large number of augmented tampered images. We consider two schemes of inpainting - simple inpainting and semantic inpainting. Apart from inpainting, we also perform feathering [23], a form of image composition. We describe in detail the approach involved in each of these schemes below.

1) **Simple Inpainting**: In this scheme, pixels belonging to a *small* region of the image are set to zero. A complete image is obtained using inpainting algorithms like [22][24]. This forms a tampered image with inpainting providing artefacts of tampering. It is also made sure that the randomly selected region does not contain more than 30% of the total number of pixels in the image. This ensures that the inpainting algorithm used to get the complete image from the tampered region provides a good performance.
2) **Semantic Inpainting**: Here, instead of a small region, an object or other semantic concept is chosen to inpaint. This requires that the ground truth mask or bounding box for the object or semantic concept is available for tampering. Images from MS COCO dataset [11] which come along with segmentation masks are employed for inpainting. If multiple semantic concepts are available with their corresponding segmentation masks in a single image, one of them is chosen at random to inpaint. This helps to augment the 'object removal' form of tampering.
3) **Feathering**: In this setting, instead of inpainting the selected object of interest, we feather (equivalent to a splicing operation) it onto another image which mainly contains background region. Let F be an image containing the foreground object/semantic concept of interest and α be the mask of the object. Given a background image B, the feathering is carried out as $I = \alpha \cdot F + (1 - \alpha) \cdot B$. This feathering is similar to the ones described in [23][25]. The background image is resized to the size of foreground image in order to feather the image. We use the foreground images provided by [25]. The background image belongs to images from COCO [11] database. A poisson blending operation [26] is also performed in order to remove any visually distinct cues at the boundary regions [18].

For all the above schemes, images are sampled from MS COCO dataset [11] as this dataset has a good balance of different objects under different scenes and lighting conditions. All the above three schemes present a fast and straightforward way to generate the tamperings. Example of each type of these tamperings is shown in Figure 2. It is to be noted here that the algorithmically generated tamperings may not showcase image-level realism, i.e the generated tamperings may not be realistic for some images. We employ a patch based approach for training the network, and hence global level coherency in terms of realism is not a necessity. Moreover, as we are interested in the local artefacts that are generated and not on the extent of realism, these images still serve as good source of augmenting the training data. We generate about 52,000 images using the tampering methods as described above. Along with 50,000 other pristine images drawn from the same dataset, the size of the algorithmically generated data is about 102,000 images.

B. Domain Adaptation

Although the generated data provides ample number of training samples, the cues offered by the artefacts produced using this approach may not be exactly similar to the cues offered by the artefacts produced manually using tampering softwares. In addition, the algorithm employed to perform tampering using a tampering software may be different from the ones used for augmenting the data. Hence, these tampered images might contain slightly different artefacts. Therefore, the generated inpainted data and the one obtained manually potentially belong to different domains. It becomes important to take into account this domain discrepancy. We perform domain adaptation between these two forms of data using a Maximum Mean Discrepancy (MMD) measure [12]. We describe briefly the approach involved and the architecture used.

The algorithmically generated data for augmentation forms the source data $X_s = \{x_s^i\}_{i=1}^{N_s}$ with abundant labelled training examples. The images from one of the datasets like CASIA TIDE V2 [8] or IEEE Forensics [10] forms the target domain data $X_t = \{x_t^i\}_{i=1}^{N_t}$ which has less number of training samples. The setup is a supervised domain adaptation task in which labels for both source domain and the target domain is available. The overview of the network employed is given in Figure 3. The network is a siamese network with a separate classification head for the two forms of data. One branch of the network takes input from the source domain while the other takes the input from the target domain. It has five convolutional layers interleaved by max pooling and ReLU activation, similar to the architecture of [5]. The size of the input is image patches of size $64 \times 64 \times 3$. The initial five convolutional layers and the first fully connected layer of size 512 learn a shared representation between the source and the target domains. Each domain is then passed through an adaptation layer which is a fully connected layer of size 256. This follows closely from [12]. A Maximum Mean Discrepancy (MMD) loss is performed on the output of this layer in order to bring the embeddings of these two domains closer. Further, the output of the adaptation layer is also passed through a fully connnected layer followed by a softmax activation. The final fully connected layer serves for classification. All the layers use ReLU activation except for the last layer, which uses softmax.

Let the set of image patches obtained from source domain data X_s be $\hat{X}_s = \{\hat{x}_s^i\}_{i=1}^{N_s}$ and the target domain X_t be $\hat{X}_t = \{\hat{x}_t^i\}_{i=1}^{\hat{N}_t}$. MMD loss is defined as :

$$\mathcal{L}_M = \min \left\| \frac{1}{|\hat{X}_s|} \sum_{\hat{x}_s \in \hat{X}_s} \Phi(\hat{x}_s) - \frac{1}{|\hat{X}_t|} \sum_{\hat{x}_t \in \hat{X}_t} \Phi(\hat{x}_t) \right\|_2^2 \tag{1}$$

where $\Phi(\hat{x}_s)$ is the output of the source adaptation layer on the input source image patch $\hat{x}_s$ and $\Phi(\hat{x}_t)$ is the output of the target adaptation layer on the input target image patch $\hat{x}_t$.

Along with the above MMD loss, normal cross entropy classification loss on the output of the softmax layer from each head is also used to classify the source and target samples.

$$\mathcal{L}_c = \min -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{2} \mathbb{1}(y^i = j) \log(p_{ij}) \tag{2}$$

where $y^i \in \{0, 1\}$ is the label corresponding to the patch sample i and $\mathbb{1}(\cdot)$ is an indicator function. p_{ij} is the probability that a patch i belongs to class j as output by the final softmax layer of the network. The overall loss is:

$$\mathcal{L} = \mathcal{L}_{ct} + \lambda_1 \mathcal{L}_{cs} + \lambda_2 \mathcal{L}_M \tag{3}$$

$\mathcal{L}_{ct}$ and $\mathcal{L}_{cs}$ are independent classification losses (Equation 2) on the target domain and the source domain input data respectively. λ_1 and λ_2 are the hyper-parameters. Here, λ_2 also controls the amount of confusion to be created between the domains. The network is trained by employing a stochastic gradient descent optimization procedure. Once the network is trained, a forward pass can be made through the target domain branch of the network to make the inference.

IV. Experiments

A. Datasets and Experimental settings

The proposed approach is tested on two standard datasets available for image tampering detection: IEEE Forensics Challenge [10] and CASIA TIDE V2 [8]. The manipulation examples are usually generated by copy-clone, splicing and removal techniques.

IEEE Forensics [10] contains 1050 pristine images and 451 tampered images. For this dataset, we have used only the publicly available training set. CASIA [8] contains 7491 pristine and 5123 tampered images. For CASIA dataset, it was observed in [27] that the tampered images were saved twice in jpeg format, in comparison with that of pristine images, which were saved only once. This resulted in the network developing bias towards the artefacts of double-jpeg compression rather than that of tampering. In order to alleviate this effect, we employ the same procedure as performed in [27]. Since there is no common available split for both the datasets, we divide the whole data into three random subsets - training (75%), validation (10%) and testing (15%). The split is made such that every split has almost equal number of copy-clone, splicing and removal tamperings among all the images of that split.

Patches of images of size $64 \times 64 \times 3$ are used to train the network. During training, the patches are sampled randomly from each image. If the selected image sample is a pristine image, a randomly sampled patch of the required size is chosen. If the image selected is tampered, then a patch is randomly sampled such that at least 10% of the pixels of the patch are tampered. We make use of the tampering ground truth mask to sample in this manner. For CASIA dataset, soft ground truth mask is generated as described in [2]. This random sampling is done for every mini-batch independently. During testing, patches are exhaustively extracted from each test image. Each of these patches are forward-passed through the network and patch-level classification is made. The image is classified as tampered if at least k patches are classified as tampered by the network. Here k is a hyper-parameter to be chosen on the validation data.

Stochastic gradient descent with momentum is used for optimizing the network parameters with an initial learning rate of 0.01 and weight decay factor of 0.0005. The value of momentum is set to 0.9. For classification, accuracy in percentage is used to report the results. In addition, precision, recall and F-score for tampering detection is also reported for the domain adaptation setting. Precision is defined as the ratio of number of true positives by the total number of true positives and false positives. Recall is the ratio of number of true positives by the total number of true positives and false

	IEEE Forensics	CASIA
Rota et al. [2]	83.24	72.29
Bappy et al. [3]	86.75	75.84
Train from scratch	82.25	71.61
Finetune on generated data	87.62	74.75
Generated data + MMD	**89.12**	**77.43**

TABLE I: Results of image tampering detection (accuracy in %) on the two datasets using different methods. The results for [3] and [2] are obtained on our setting after reproducing the result on the original setting of the paper.

	Precision	Recall	F-score
IEEE Forensics	0.944	0.766	0.846
CASIA	0.814	0.704	0.755

TABLE II: Precision, Recall and F-score on different datasets on the domain adaptation setting.

negatives. F-score is the harmonic mean between precision and recall.

In order to better understand the source of performance of the proposed approach, various control experiments (baselines) are also performed. The various control setups can be described as follows:

- Train from scratch: The whole network is trained from scratch without using the inpainted and composited data. This denotes the performance that the network can achieve when trained without using algorithmically generated data.
- Finetune on generated data: The network is trained after initializing with weights trained on inpainted and composited data.

For the baselines, the architecture which corresponds to one of the streams of the siamese network in Figure 3 is used.

B. Results

The results for image tampering detection are tabulated in Table I. It can be seen that training from scratch leads to comparatively lower results. This might indicate that the amount of data required to train a large network from scratch is not sufficient with just the data from the datasets available publicly. This hypothesis is further supported by the third baseline which improves the detection rates on all the datasets with respect to training from scratch. It also indicates that the network trained on the algorithmically generated data is well transferable to image tampering detection. On performing domain adaptation, it was observed that there was a non-trivial increase in accuracy on both the datasets. This indicates that the algorithmically generated tamperings and the tamperings obtained using proprietary softwares have slightly different distributions. However, on performing MMD, the domain shift problem is alleviated to a certain extent.

C. Ablation Studies

In order to better understand the degree of effect of performance of the generated data, we vary the number of algorithmically generated images used for augmentation and

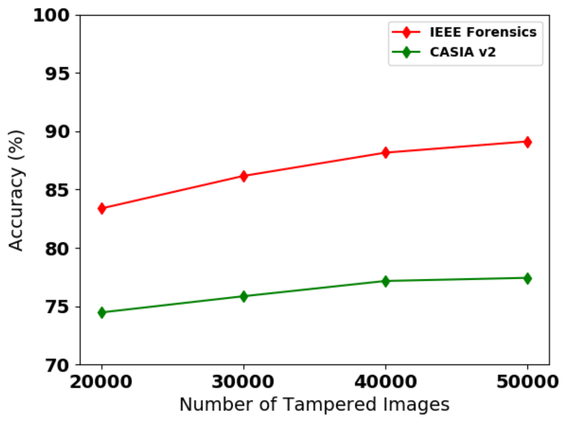

Fig. 4: Plot of number of tampered images in the augmented data against detection accuracy in %.

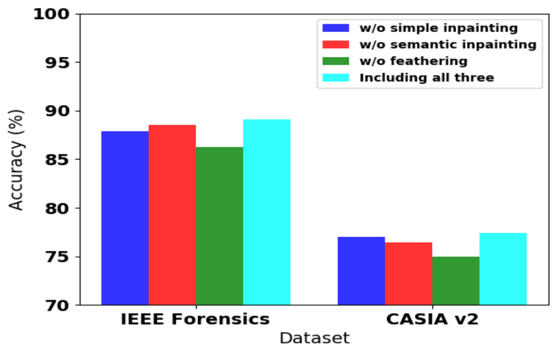

Fig. 5: Plot indicating the performance of the proposed approach on tampering detection due to various schemes of data augmentation. The first bar (sky blue) corresponds to performance of the algorithm when the augmented data does not consist of simple inpainting type of tampering. The second bar (blue) corresponds to the performance of the algorithm when the augmented data does not consist of semantic inpainting type of tampering. The third bar (yellow) denotes the performance of the algorithm when the augmented data does not consist of feathering type of tampering. The fourth bar (green) indicates the performance due to all the three types of schemes.

observe the change in performance. The corresponding graph is shown in Figure 4. It is made sure that the number of tampered images and the pristine images of the augmented data is maintained at a ratio of 1:1. It is also made sure that for generating tampered images, the number of images generated using each of the schemes are nearly equal. It can be seen that as the number of tampered images used for augmentation goes up, the detection accuracy also grows, indicating that augmenting data is beneficial for learning good representations of tamperings. As we augment more data, the gain in performance is higher for IEEE dataset than that for CASIA. This is expected as the number of samples in IEEE is much lesser compared to the CASIA.

In addition, we also attempt to isolate the performance due to each of the augmentation schemes described in Section III-A. Figure 5 presents the results on various settings. For each setting, the total number of tampered images is equal to the one used for the proposed approach, i.e around 50,000. It can be seen that drop in accuracy is highest when feathering form of tampering is excluded. This indicates that feathering contributes the most among all the schemes. The other schemes offer varying levels of gain in performance on the two datasets. Note that the performance of each of these setups is still higher than the performance when the network is trained from scratch (Table I) indicating the positive contribution of each of the proposed schemes.

D. *Comparison with the state-of-the-art*

In order to demonstrate the competitiveness of the proposed approach, we provide comparison with the state-of-the-art. We implement the approach of [3] and [2] as they have been shown to provide state-of-the-art for image tampering detection. For [3], we verify that our reproduced results are close to the reported result for the task of patch classification accuracy on the IEEE dataset. Similarly, we reproduce the result of [2] on the CASIA TIDE V2 dataset. We test the approaches of both [3] and [2] on our setting as described earlier. The results are tabulated in Table I. The proposed approach achieves superior accuracy, albeit being simpler in architecture. This demonstrates the efficacy of the data augmentation method and the importance of more data for image tampering detection. In addition, we would also like to point that the proposed data augmentation method is relevant in any other CNN based approach. Therefore, the proposed augmentation method is also potentially beneficial in the setting of [3] as well as [2], wherein the detection accuracies can be even higher.

V. Conclusion

In this work, we present an efficient data augmentation method for image tampering detection. We employ inpainting and compositing schemes to algorithmically create a dataset which contains artefacts similar to the ones produced using proprietary softwares. We consider three schemes for augmenting - simple inpainting, semantic inpainting and feathering in order to generate the data. In combination with a state-of-the-art domain adaptation method, we demonstrate that the proposed method of augmentation is effective in terms of increasing the tampering detection accuracies on two standard datasets. This experimental result opens up the possibility of training an end-to-end model using a generative adversarial network [28] wherein the generator produces the inpainted data which can augment the smaller datasets produced by proprietary softwares, while jointly training a discriminator which can detect tampered images. This line of work forms a part of our future work. Augmentation of data to learn efficient representations of tamperings for tamperings apart from copy-paste, removal and splicing is also an interesting and important direction in order to develop models that can detect tampered images in the wild.

References

[1] T. Bianchi and A. Piva, “Image forgery localization via block-grained analysis of jpeg artifacts,” *TIFS*, 2012.

[2] P. Rota, E. Sangineto, V. Conotter, and C. Pramerdorfer, “Bad teacher or unruly student: Can deep learning say something in image forensics analysis?” in *ICPR*, 2016.

[3] J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. Manjunath, “Exploiting spatial structure for localizing manipulated image regions,” in *ICCV*, 2017.

[4] S. Dadkhah, A. A. Manaf, Y. Hori, A. E. Hassanien, and S. Sadeghi, “An effective svd-based image tampering detection and self-recovery using active watermarking,” *Signal Processing: Image Communication*, 2014.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NIPS*, 2012.

[6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016.

[7] J. Chen, X. Kang, Y. Liu, and Z. J. Wang, “Median filtering forensics based on convolutional neural networks,” *SPL*, 2015.

[8] “CASIA TIDE v2,” http://forensics.idealtest.org/casiav2/.

[9] B. Bayar and M. C. Stamm, “A deep learning approach to universal image manipulation detection using a new convolutional layer,” in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, 2016.

[10] “IEEE IFS-TC Image Forensics Challenge Dataset.” http://ifc.recod.ic.unicamp.br/fc.website/index.py.

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *ECCV*, 2014.

[12] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, “Deep domain confusion: Maximizing for domain invariance,” *arXiv preprint arXiv:1412.3474*, 2014.

[13] Y. Rao and J. Ni, “A deep learning approach to detection of splicing and copy-move forgeries in images,” in *WIFS*, 2016.

[14] J. Fridrich and J. Kodovsky, “Rich models for steganalysis of digital images,” *TIFS*, 2012.

[15] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *AISTATS*, 2010.

[16] L. Bondi, S. Lameri, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, “Tampering detection and localization through clustering of camera-based cnn features,” in *CVPR Workshops*, 2017.

[17] Q. Chen, J. Huang, R. Feris, L. M. Brown, J. Dong, and S. Yan, “Deep domain adaptation for describing people based on fine-grained clothing attributes,” in *CVPR*, 2015.

[18] D. Dwibedi, I. Misra, and M. Hebert, “Cut, paste and learn: Surprisingly easy synthesis for instance detection,” in *ICCV*, 2017.

[19] P. P. Busto, J. Liebelt, and J. Gall, “Adaptation of synthetic data for coarse-to-fine viewpoint refinement.” in *BMVC*, 2015.

[20] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, “The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes,” in *CVPR*, 2016.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *IJCV*, 2015.

[22] A. Criminisi, P. Pérez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” *TIP*, 2004.

[23] J.-Y. Zhu, P. Krahenbuhl, E. Shechtman, and A. A. Efros, “Learning a discriminative model for the perception of realism in composite images,” in *ICCV*, 2015.

[24] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, “High-resolution image inpainting using multi-scale neural patch synthesis,” in *CVPR*, 2017.

[25] N. Xu, B. Price, S. Cohen, and T. Huang, “Deep image matting,” in *CVPR*, 2017.

[26] P. Pérez, M. Gangnet, and A. Blake, “Poisson image editing,” *TOG*, 2003.

[27] P. Sutthiwan, Y. Q. Shi, H. Zhao, T.-T. Ng, and W. Su, “Markovian rake transform for digital image tampering detection,” in *Transactions on data hiding and multimedia security VI*, 2011.

[28] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *NIPS*, 2014.