DC-NET: Discriminate Counterfeit Network For Forgery Detection Through Compressed Information and Attention Mechanism

Chih-Jung Chang*[1], Chi-Luen Feng*[1], Shun-Ta Wang*[2], and Jun-Cheng Chen[3]

[1]National Taiwan University
[2]National Taiwan Normal University
[3]Research Center for Information Technology Innovation, Academia Sinica

Abstract

Nowadays real-life image forgery has become a serious issue that has huge impacts on every aspect of our lives. As a result, we have witnessed increasing interests in studying forgery detection mechanisms. Recent research [32] has seen the forgery localization problem of an image as a local anomaly detection problem. Based on this idea, we discuss several network structures to realize local anomaly detection of an image and compare the performances of these different structures. The progress made by this study will provide some new ideas to further improve the overall performance of image forgery detection.

1. Introduction

By the rapid progress in computer vision technology, the technique of image forgery is gradually beyond control. Such as face swapping [16], deepfake [28], video manipulation, most of these newest methods gradually erode our daily life. Moreover, because of the appearance of the Deep Neural Network, we can hardly identify whether an image is real or not. The consequence is that we are susceptible to the manipulation of false information. Therefore, defending these forgery information is the top topic for human beings.

Image forgery includes lots of methods. Among all, copy-move [23, 6, 31], splicing [7, 13, 33], removal [34] and enhancement [2, 1, 5] are the four that have usually been studied. Copy-move forgery is a process of cutting an object from one image, then paste this object on the same image after doing rotation or reshape. Splicing forgery, similar to the copy-move forgery, is a process of making a composite picture by cutting some object from one image and adding it to the other one. The main difference between copy-move forgery and splicing forgery is that splicing forgery is more difficult to detect. Because it's easy to detect similar objects in the same image, whereas it's hard to identify different objects with different image features. As for removal forgery, it's a process of removing some objects from the original picture and using the background information to bring back the empty part. Finally, enhancement is the process of adjusting image features, including adjusting the white balance value, or blurring part of the picture, etc.

However, one image may contain numerous forgery methods in the real world. For example, after using copy-move, enhancement can be performed on the forgery boundary, making it harder to be detected as a forgery image. Besides, take advantage of the popularity of commercial software(ex: Adobe Photoshop), it's possible to make a forgery picture without any professional knowledge. Under such circumstances, we desperately need a model that can tell us whether the current photo has been tampered with.

In this paper, we address these issues by developing a model which can localize the forgery region and preserve the integrity of the entire picture. By detecting image characteristics for each pixel, our proposal model is applicable to any forgery method. We adopt two different methods to construct our models. The first is to use the 2-stream method, the main stream of the model detects the anomaly elements in the picture to identify the forgery region in the image; another stream is to establish a coarse map for guiding the model to predict a more robust answer. The second method is to use the attention method to calculate the similarity of each pixel to the pixel of the entire picture. This method finds the pixels with the lowest relationship and regards them as forgery regions.

The remainder of the paper is organized as follows. Sec.2 discusses the related work. Sec.3 talks about our model practice process, including how to do the pre-

*Equal contribution

processing of the dataset, the construction details of different models, and discuss the performance differences caused by different object functions. Sec.4 mentions the experiment result for our models. Sec.5 summarizes the results of this work and draws conclusions.

2. Related Work

2.1. Image Manipulation Methods

In addition to the four most commonly studied image manipulation techniques (*i.e*., copy-move, splicing, removal, and enhancement). There are some post-processing methods like resizing, rotation or translation to make the tampered area better blend with the background. Besides, with the evolution of deep neural network(DNN) technology, many DNN methods have been used for entire or partial image generation, such as AutoEncoders (VAEs) [15, 24] and Generative Adversarial Networks (GANs) [12]. The term Deepfake has become a synonym for DNN based face manipulation methods. There are so many DeeFake applications in our daily life, the most famous one is FaceApp [10]. FaceApp can select the facial attributes you want to modify, synthesising an entire new face.

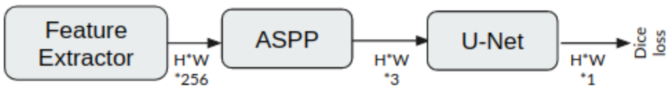
2.2. Manipulation Localization

Image manipulation localization can be regarded as a local anomaly detection problem. The goal is to detect the locations which may be tampered with. There are many DNN-based methods to deal with this problem. Face X-Ray [17] focus on the blending boundary for a forged image and the absence of blending for a real image. Qian *et al*. [22] find some subtle forgery artifacts in frequency domain, taking advantage of two different but complementary frequency-aware clues to do forgery detection. Some other works [8, 3] exploit attention mechanisms as a guide to detect the forged location. Wang and Dantcheva [30] use the 3DCNN model in detecting manipulated videos. Nguyen *et al*. [21] see this problem as a multi-task segmentation problem. They proposed a VAEs model to simultaneously detect manipulated images and videos and locate the manipulated regions. ManTra-Net [32] utilizes local Z-score features and puts them into a long short-term memory to detect local anomalies. However, these DNN based techniques do not take local anomaly features as a guiding mask and consider features from global to local at the same time, which would miss some intrinsic features of forgery images.

3. Methods

3.1. Datasets

In order to enable the model to cope with a variety of forgery conditions, we follow the way Mantra-Net constructs the datasets. We use Dresden [11] and COCO [18] as our datasets, and pre-process the original images through copy-move, splicing, removal and enhancement. The following will explain one by one how we use the above forgery methods to process the datasets.



(a) ASPP + U-Net

Figure 1. (a) Using the first half of the MantraNet, then concat ASPP and U-Net

Copy-move and splicing both use the annotations that come with COCO dataset, which helps us to mark the object in the picture. In the process of copy-move, we cut the largest object of a photo, rotate and resize it at random angles and random shape, and paste it anywhere in the picture. As for the process of splicing, we first randomly take two photos in the COCO dataset, then select the object with the largest area of one of the photos as the splicing object, and after the same random angle of rotate and resize, the splicing object paste on the remaining images.

Removal and enhancement use Dresden dataset, first we follow [19], introducing random structured binary mask M. In the process of removal, we dilate the binary mask M, rotate and resize it at random angles and random shape first, then we overlap Dresden's picture with binary mask M. The region which is masked by binary mask M are set to zero, and fill this region with the background. As for the process of enhancement, we overlap Dresden's picture with binary mask M, then do some enhancement process on the masked region, which includes blurring, morphological operations, noise, quantization, auto contrast, equalization, and compression.

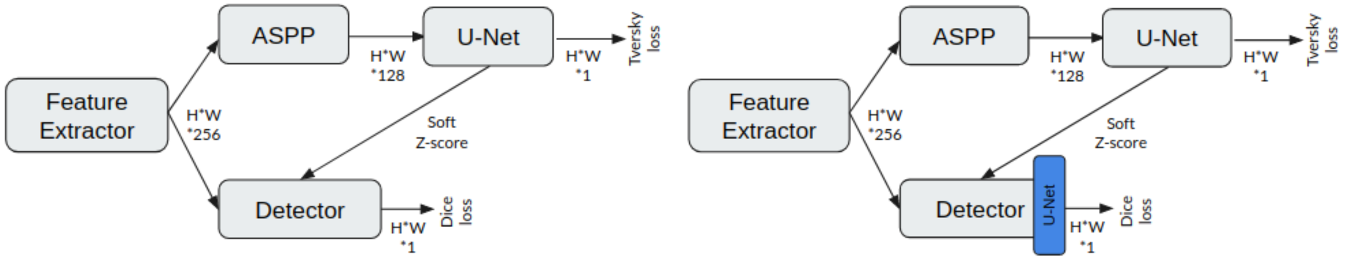
After doing the pre-processing above, the images are cropped to 256×256, and these images are our training and validation dataset.

3.2. Network Structures

Given a feature map, how does one identify potential forged regions? Inspired by ManTra-Net, we turn this forgery localization problem into a local anomaly detection problem. That is, any pixel with sufficiently different features from the dominant feature of an image is very likely to be forged. In order to spot anomalies in a given image, correlation between the pixels must be considered. In this section, we propose three methods that aim to compute the correlation between the pixels and utilize such information to classify whether or not a pixel is forged.

3.2.1 ASPP + U-Net

Atrous Spatial Pyramid Pooling (ASPP) [4] is a common method for image segmentation, which adopts different



(a) Guided Z-score

(b) Adding U-Net to the detector

Figure 2. (a) Using the coarse mask generated by the ASPP + U-Net module as a guide to compute Z-scores (b) Substituting the decision layer in the detector for U-Net to improve the quality of the prediction masks

sizes of dilated convolution filters to capture information in different scales. It provides an effective way to control the size of the receptive field, making the model to strike a balance between local and global features.

U-Net model was first proposed by Ronneberger *et al*. [25], which has been widely used in medical image segmentation tasks. U-Net consists of two paths, the contracting path and the expanding path. The contracting path is mainly used to capture the contextual information in the image, and the following path is the expanding path which adopts a fully convolutional network to do upsample so as to achieve precise localization. In addition, the contracting path is combined with the upsampled output, considering more contextual information.

In this work, we use a VGG16 [27] architecture pretrained by ManTra-Net as a manipulation trace feature extractor, which was trained as a manipulation classification model excluding the decision block. The classification model was trained for 385 fine-grained image manipulation classes, so the extracted feature is really robust and sensitive. Then we concat ASPP with U-Net behind the VGG16, the reason using these two structures is that we want to boarden model's receptive field in ASPP structure, and utilize all the information we learnt in U-Net structure. ((see Fig. 1))

3.2.2 Guided Z-score

As suggested by Wu *et al*. [32], Z-score is also a way to quantify how the feature of a pixel is different from that of the rest of an image. Let F be a raw feature tensor of size $C \times H \times W$. One can compute

$$Z_F[i,j] = \frac{F[i,j] - \mu_F}{\sigma_F} \tag{1}$$

where μ_F is the average feature of an image

$$\mu_F = \sum_{i=1}^{H} \sum_{j=1}^{W} F[i,j]/(HW) \tag{2}$$

and σ_F is the standard deviation of F

$$\sigma_F^2 = \sum_{i=1}^{H} \sum_{j=1}^{W} F[i,j]^2/(HW) - \mu_F^2 \tag{3}$$

Z_F defined in Eq.(1) encodes how different each pixel is from the dominant feature of an image. To deal with the drawback when more than two regions in the same image are forged, which makes μ_F less likely to represent the dominant pristine background feature, Wu *et al*. suggest computing the window-wise deviation feature and collecting a series of such features w.r.t. different window sizes,

$$Z_F^{n\times n}[i,j] = \frac{F[i,j] - \mu_F^{n\times n}}{\sigma_F} \tag{4}$$

where $\mu_F^{n\times n}$ is the average feature computed within the $n \times n$ window centered at (i,j) location of an image.

The sequence of Z-score features

$$Z_F^* = [Z_F^{n_1\times n_1}, \ldots, Z_F^{n_k\times n_k}, Z_F] \tag{5}$$

is then concatenated along a new time dimension and fed into a convolutional LSTM layer to mimic the coarse-to-fine human decision progress.

However, the window-wise average feature $\mu_F^{n\times n}$ still suffers from being contaminated by the feature of the forged region within the window. We thus propose the following Guided Z-score to collect only the pristine background information even better.

Suppose we have a mask that coarsely predicts the forged regions of an image, we can then exclude these potentially forged regions when computing the average feature μ_F and $\mu_F^{n\times n}$. The aforementioned ASPP + U-Net structure serves as a good mask generator. Concretely, denote the prediction from the U-Net by P, we can obtain the binary mask M by thresholding P as shown in Eq.(6).

$$M[i,j] = \begin{cases} 1, & \text{if } P[i,j] \leq \xi \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

for some ξ between 0 and 1. Note that M is inverted in the way that the pixels from the pristine background are of value 1 and the potentially forged pixels are of value 0.

We can then compute μ_F and σ_F as shown in Eq.(7) and (8).

$$\mu_F = \sum_{i=1}^{H}\sum_{j=1}^{W}(M[i,j]F[i,j])/S_M \tag{7}$$

$$\sigma_F^2 = \sum_{i=1}^{H}\sum_{j=1}^{W}(M[i,j]F[i,j])^2/S_M - \mu_F^2 \tag{8}$$

where S_M is the total number of background pixels in an image

$$S_M = \sum_{i=1}^{H}\sum_{j=1}^{W}M[i,j] \tag{9}$$

One drawback of the design above is that the thresholding function cannot be back-propagated, which disables the possibility of end-to-end training. We thus propose another way to compute the soft mask M':

$$M' = softmax(-P) \tag{10}$$

Here M' can be interpreted as a distribution. Each value of M' suggests how likely the corresponding pixel is to come from the pristine background. In this situation μ_F and σ_F are computed as follows:

$$\mu_F = \sum_{i=1}^{H}\sum_{j=1}^{W}M'[i,j]F[i,j] \tag{11}$$

$$\sigma_F^2 = \sum_{i=1}^{H}\sum_{j=1}^{W}M[i,j]F[i,j]^2 - \mu_F^2 \tag{12}$$

By leveraging the coarse mask to compute more precise μ_F and σ_F, the model is able to spot forged regions in an image more accurately. One can refer to Sec.4 to see how this Guided Z-score method affects prediction.

3.2.3 Non-Local

In order to utilize the correlation between "all the pixels" in the image, we try the network called Non-local Network [29]. By comparison, a traditional convolution network only can see the correlation in the given kernel size. However, a non-local network can use the attention mechanism to see the whole image information:

$$y_i = \frac{1}{HW}\sum_{\forall j} f(x_i, x_j)g(x_j), \tag{13}$$

where $f(x_i, x_j)$ is the correlation function between a pair of pixels, and $g(x_j)$ is a representation of itself.

We use the first half of the ManTra-Net as the backbone, and then concatenate the non-local network to the back. The purpose of this structure is to make the model consider the information of all pixels. When using this structure, we believe that the lower the similarity to the entire image, the more likely it is an anomaly pixel. So by using this method, the model can compare which pixel has a problem from the pixel level, so it isn't limited to which forgery method.

However, we found a problem during training that non-local network need a huge amount of memory, it's hard to use a larger batch size in order to lower the variance in the data, so we adapt two methods to deal with this problem. One is to do downsampling according to Zhu *et al.* [35], we use pooling to reduce the dimension of non-local network when calculating key and value, dimension is reduced from $H \times W \times H \times W$ to $H \times W \times 110$ (see Fig. 3(a)). We are equivalently saving nearly 595 times of space in this calculation if the image size is 256×256. It allows us to increase our batch size for stable training.

Another method is to use an idea similar to U-Net. First we reduce the dimension of height and width, then go through the non-local network and do upsampling to restore the original dimension (see Fig. 3(b)). Using the above method to replace the original non-local block can achieve the goal of saving memory space.

By using a non-local network, our model can break through the limitations of the traditional CNN's receptive field, so as to compare each pixel, and finally find the pixel with the lowest correlation and treat it as anomaly pixels. More experiment results can be found at Sec.4.

3.3. Objective Functions

Image manipulation detection can be regarded as a binary image segmentation task. For each pixel, we identify whether it is fake or not. Therefore, we use two common image segmentation loss functions as our objective functions.

Dice Loss [20]: Eq.(14) is the Dice score coefficient, which is commonly used in semantic segmentation tasks. It calculates the proportion of the overlap between the predicted mask and the ground-truth mask.

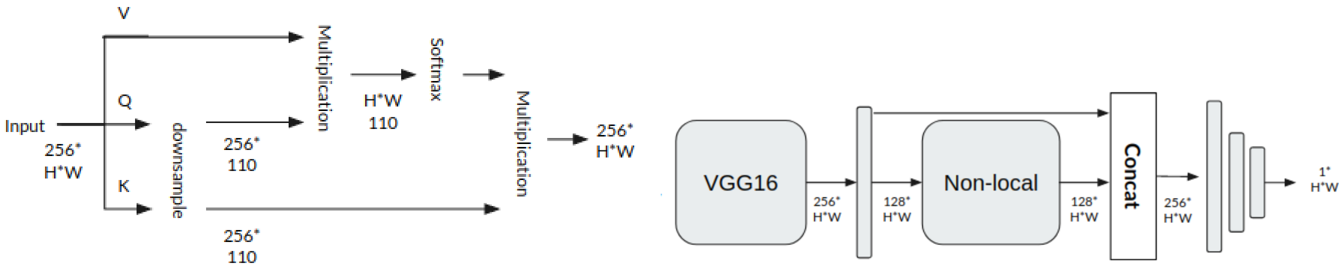
$$Dice = 2\frac{|A \cap B|}{|A| + |B|} \tag{14}$$

where $|A \cap B|$ is the intersection area of the predicted mask and the ground-truth mask. $|A|$ is the number of pixels in the predicted mask, and $|B|$ is the number of pixels in the ground-truth mask.

$$DiceLoss = 1 - Dice \tag{15}$$

where the Dice Loss is one minus the Dice score coefficient.

Tversky Loss [26]: Since Diceloss has the same weight



(a) Downsample

(b) U-Net like

Figure 3. (a) The downsample method for Non-local Network (b) The U-Net like Non-local Network

when calculating FPs and FNs (precision and recall). For an ideal coarse mask, lack of FPs is an acceptable trade-off for higher FNs. In order to weigh FNs more than FPs in training our network, we adopt Tversky loss as another objective function. The Tversky index is defined as:

$$T(A,B) = \frac{|A \cap B|}{|A \cap B| + \alpha|A-B| + \beta|B-A|} \tag{16}$$

where α and β are the magnitude of penalties for FPs and FNs respectively.

$$TverskyLoss = 1 - T(A,B) \tag{17}$$

where the Tversky Loss is one minus the Tversky index.

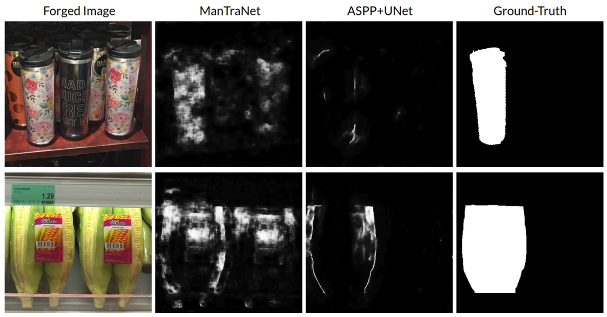


Figure 4. Visual results on COVERAGE [31] dataset. From left to right: the forged image, the manipulated area prediction by ManTra-Net, the manipulated area prediction by our ASPP+UNet model, and the ground-truth.

4. Experiments and Discussion

We have previously discussed three different methods to perform the image forgery detection task. In this section, we provide both quantitative and qualitative results to compare the performances of the three methods.The three benchmark datasets we use for testing are CASIA [9], COVERAGE [31], and Columbia dataset [14].

Regarding evaluation metrics, we use the pixel-level area under the receiver operating characteristic curve (AUC of ROC) as used in [32]. However, we found that AUC might not be the best way to evaluate the performance of a model and we will discuss this issue later in this section. Therefore, we strongly recommend one to take both quantitative and qualitative results into consideration when it comes to evaluating the performance.

	COVERAGE	CASIA	Columbia
ASPP + U-Net	78.6	68.9	81.2
Guided Z-score	68.7	63.2	70.3
Guided Z-score w/ U-Net	76.7	65.5	68.8
Non-Local	63.9	56.7	58.2
ManTra-Net	85.4	77.1	86.8

Table 1. Performance comparisons. The average AUC on COVERAGE, CASIA, and Columbia dataset

In terms of general training settings, we set batch size to 4 and use apex for mixed precision training. We use the Adam optimizer without decay and the learning rate will be halved if validation loss fails to improve over 10 epochs.

4.1. ASPP + U-Net

Table 1 shows the quantitative results, We still have a gap of about 7% average AUC between our ASPP + U-Net model and the ManTra-Net's performance. But as for the qualitative results (see Fig.4), it shows that AUC couldn't truly reflect the manipulation detection accuracy of the model. Especially in some copy-move cases, it makes sense that the detector could only detect the outline instead of the whole tempered area, because there is no essential difference between the area inside the line and another parts of the image. When human eyes see the outline, it can naturally determine which object is manipulated. However, in this case, the outputs of our ASPP + U-Net model get rather lower AUC than theirs, but our method is better than them in depicting the outline. In their results, it could be difficult for human eyes to recognize which one is the tampered object.

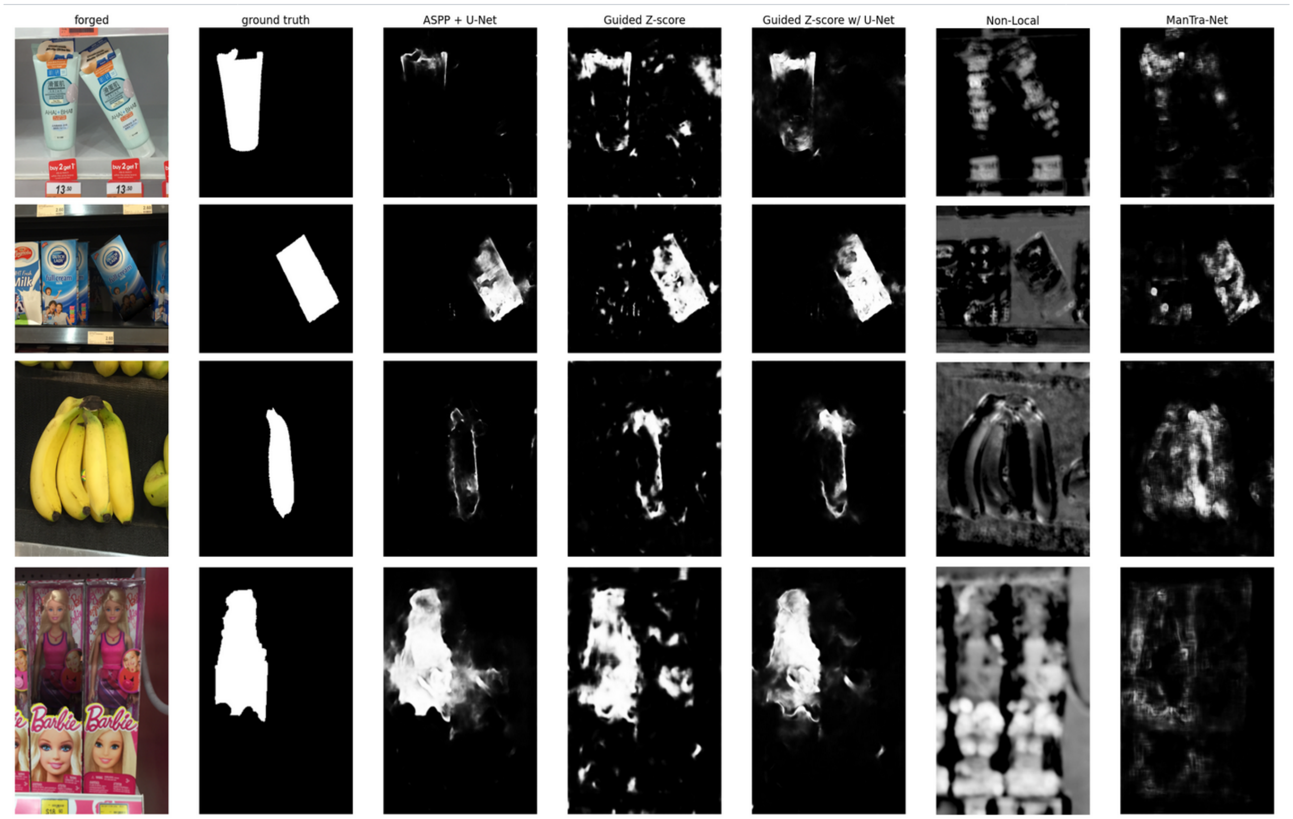


Figure 5. Visual comparisons on COVERAGE [31] dataset. The first two columns are the forged images and the ground-truth images. The 3rd to 6th columns show the results using our proposed methods. The last column is the prediction by the ManTra-Net [32]

4.2. Guided Z-score

To see how using the coarse mask as a guide to compute z-score affects the prediction, we follow the structure of the ManTra-Net [32] and add a new branch of ASPP + U-Net module to allow computing z-score by our proposed method. The whole architecture can be found in Fig.2a.

As shown in Fig.6, the ASPP + U-Net branch does provide decent coarse masks for the forgery prediction. The model first classifies the pixels of an image into three classes: the pristine background, the forged region, and the suspicious area (*i.e.*, the black, white, and gray part of an output coarse mask). The model is then able to output the more accurate and refined prediction using our proposed method to compute the guided Z-score.

Although the average AUC are still lower than the original method proposed by Wu *et al.* [32] on all the three datasets, we do see some improvements on several images (See Fig.5). Our method generally provides more confident prediction and is able to reduce the false positives in an image. We further improve the quality of the prediction masks by substituting the decision layer in the ManTra-net for the U-Net structure (see Fig.2b), which results in sharper edges and less noise in the final outputs (one can compare the 4th and the 5th columns in Fig.5 to see the difference).

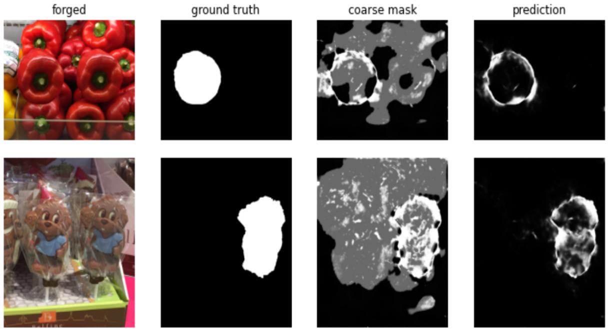


Figure 6. Outputs from the model using the Guided Z-score method. The third column from the left shows the coarse masks generated by the ASPP + U-Net module, and the last column is the final predictions made by the model.

4.3. Non-Local

In this model setting, we replace the second half of Mantranet with a Non-local network, hence the model structure is VGG16 + Non-local network. Besides, we tried two different versions for non-local network. One is to reduce the dimensionality which is similar to U-Net before passing to non-local block, and the other is to do downsample with reference to Zhu *et al*. [35]. The purpose of these two versions is to allow the model to attend to the information of the entire photo.

In the experiment setting, we followed Mantranet to use a pretrained VGG16. and then trained VGG16 with a non-local network simultaneously. The learning rate is set to 1e-4.

In the result, we found that it doesn't perform well[1]. The result is that if the non-local network is connected behind VGG16, it tends to enhance the value of the feature map produced by VGG16 rather than finding the correlation in the image. We can see this phenomenon in Fig. 5

In order to solve the problem mentioned above, we tried another setting method, which is to insert a non-local network into the block of VGG16, and directly use the output of the modified VGG16 as final prediction. Because the image still retains a lot of original information in the first half of the model, this setting can take more original information into consideration, help the model learn more complete information.

However, under this setting, the model only depicts the frame of the picture. So we think that the current setting of non-local network can't detect anomaly regions, changing the similarity function or model structure may be of great help.

5. Conclusion

In this paper, we introduce three different image manipulation detection ways, all of them take global and local information into account, but extract features in different ways. ASPP+U-Net architecture utilizes ASPP mechanism to learn features in different receptive fields. Guided Z-score method takes ASPP+U-Net model as a guiding mask to collect only the pristine background information, computing their Z-score to distinguish the manipulated area. Non-Local network exploits the correlation between "all the pixels", considering the pixels with the lowest correlation as anomaly pixels.

Our extensive experimental results show that, although our methods don't outperform other SOTA works in quantitative results, from the perspective of qualitative result, it shows more concise result than others, because we believe in that image manipulation detection is a work for human beings, we need to tell people where the tempered region is more straightforwardly, instead of increasing uncertainty to trade-off the highest quantitative scores like AUC, F1 Scores etc. In addition, because of the limitation of hardware environments, we couldn't train the models as other SOTA works, maybe our models would have better quantitative results in their training environments.

[1] Because the non-local network requires too much memory, we only display partial image results, not numerical results

References

[1] B. Bayar and M. C. Stamm. "Constrained Convolutional Neural Networks: A New Approach Towards General Purpose Image Manipulation Detection". In: *IEEE Transactions on Information Forensics and Security* 13.11 (2018), pp. 2691–2706.

[2] Belhassen Bayar and Matthew 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*. IH&MMSec '16. Vigo, Galicia, Spain: Association for Computing Machinery, 2016, pp. 5–10. ISBN: 9781450342902. DOI: 10.1145/2909827.2930786. URL: https://doi.org/10.1145/2909827.2930786.

[3] Diangarti Bhalang Tarianga et al. "Classification of Computer Generated and Natural Images based on Efficient Deep Convolutional Recurrent Attention Model". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2019.

[4] Liang-Chieh Chen et al. *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*. 2016. arXiv: 1606.00915 [cs.CV].

[5] H. Choi et al. "Detecting composite image manipulation based on deep neural networks". In: *2017 International Conference on Systems, Signals and Image Processing (IWSSIP)*. 2017, pp. 1–5.

[6] D. Cozzolino, G. Poggi, and L. Verdoliva. "Efficient Dense-Field Copy–Move Forgery Detection". In: *IEEE Transactions on Information Forensics and Security* 10.11 (2015), pp. 2284–2297.

[7] D. Cozzolino, G. Poggi, and L. Verdoliva. "Splicebuster: A new blind image splicing detector". In: *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*. 2015, pp. 1–6.

[8] Hao Dang et al. *On the Detection of Digital Face Manipulation*. 2019. arXiv: 1910.01717 [cs.CV].

[9] J. Dong, W. Wang, and T. Tan. "CASIA Image Tampering Detection Evaluation Database". In: *2013 IEEE China Summit and International Conference on Signal and Information Processing*. 2013, pp. 422–426.

[10] *FaceApp*. https://faceapp.com/app.

[11] Thomas Gloe and Rainer Böhme. "The Dresden Image Database for Benchmarking Digital Image Forensics." In: *J. Digital Forensic Practice* 3 (Jan. 2010), pp. 150–159.

[12] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].

[13] Minyoung Huh et al. *Fighting Fake News: Image Splice Detection via Learned Self-Consistency*. 2018. arXiv: 1805.04096 [cs.CV].

[14] *Image splicing detection evaluation dataset*. http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/photographers.htm. 2004.

[15] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2013. arXiv: 1312.6114 [stat.ML].

[16] Iryna Korshunova et al. *Fast Face-swap Using Convolutional Neural Networks*. 2016. arXiv: 1611.09577 [cs.CV].

[17] L. Li et al. "Face X-Ray for More General Face Forgery Detection". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 5000–5009.

[18] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2014. arXiv: 1405.0312 [cs.CV].

[19] Guilin Liu et al. *Image Inpainting for Irregular Holes Using Partial Convolutions*. 2018. arXiv: 1804.07723 [cs.CV].

[20] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-net: Fully convolutional neural networks for volumetric medical image segmentation". In: *2016 fourth international conference on 3D vision (3DV)*. IEEE. 2016, pp. 565–571.

[21] Huy H. Nguyen et al. *Multi-task Learning For Detecting and Segmenting Manipulated Facial Images and Videos*. 2019. arXiv: 1906.06876 [cs.CV].

[22] Yuyang Qian et al. *Thinking in Frequency: Face Forgery Detection by Mining Frequency-aware Clues*. 2020. arXiv: 2007.09355 [cs.CV].

[23] Y. Rao and J. Ni. "A deep learning approach to detection of splicing and copy-move forgeries in images". In: *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*. 2016, pp. 1–6.

[24] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. *Stochastic Backpropagation and Approximate Inference in Deep Generative Models*. 2014. arXiv: 1401.4082 [stat.ML].

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].

[26] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. *Tversky loss function for image segmentation using 3D fully convolutional deep networks*. 2017. arXiv: 1706.05721 [cs.CV].

[27] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. arXiv: 1409.1556 [cs.CV].

[28] Ruben Tolosana et al. *DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection*. 2020. arXiv: 2001.00179 [cs.CV].

[29] Xiaolong Wang et al. *Non-local Neural Networks*. 2017. arXiv: 1711.07971 [cs.CV].

[30] Yaohui Wang and Antitza Dantcheva. "A video is worth more than 1000 lies. Comparing 3DCNN approaches for detecting deepfakes". In: *FG'20, 15th IEEE International Conference on Automatic Face and Gesture Recognition, May 18-22, 2020, Buenos Aires, Argentina.* Buenos Aires, Argentina, May 2020. URL: https://hal.inria.fr/hal-02862476.

[31] B. Wen et al. "COVERAGE — A novel database for copy-move forgery detection". In: *2016 IEEE International Conference on Image Processing (ICIP)*. 2016, pp. 161–165.

[32] Y. Wu, W. AbdAlmageed, and P. 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)*. 2019, pp. 9535–9544.

[33] Yue Wu, Wael AbdAlmageed, and Prem Natarajan. *Deep Matching and Validation Network – An End-to-End Solution to Constrained Image Splicing Localization and Detection*. 2017. arXiv: 1705.09765 [cs.CV].

[34] Xinshan Zhu et al. "A deep learning approach to patch-based image inpainting forensics". In: *Signal Processing: Image Communication* 67 (June 2018). DOI: 10.1016/j.image.2018.05.015.

[35] Zhen Zhu et al. *Asymmetric Non-local Neural Networks for Semantic Segmentation*. 2019. arXiv: 1908.07678 [cs.CV].