M^3:Manipulation Mask Manufacturer for Arbitrary-Scale Super-Resolution Mask

Xinyu Yang[1], Xiaochen Ma[1], Xuekang Zhu[1], Bo Du[1], Lei Su[1], Bingkui Tong[1] Zeyu Lei[1,2]
Jizhe Zhou[1,3*]
[1]College of Computer Science, Sichuan University, China, 2021141460237@stu.scu.edu.cn
[2]Department of Computer and Information Science, University of Macao, Macao SAR
[3]Engineering Research Center of Machine Learning and Industry Intelligence, MOE, China, jzzhou@scu.edu.cn

Abstract

In the field of image manipulation localization (IML), the small quantity and poor quality of existing datasets have always been major issues. A dataset containing various types of manipulations will greatly help improve the accuracy of IML models. Images found on public forums, such as those in online image modification communities, are often manipulated using various techniques. Creating a dataset from these images can significantly enhance the diversity of manipulation types in our data. However, due to resolution and clarity issues, images obtained from the internet often contain noises, making it difficult to obtain clean masks by simply subtracting the manipulated image from the original. These noises are difficult to remove, rendering the masks unusable for IML models. Inspired by the field of change detection, we treat the original and manipulated images as changes over time for the same image and view the data generation task as a change detection task. Due to clarity issues between images, conventional change detection models perform poorly. Therefore, we introduced a super-resolution module and proposed the Manipulation Mask Manufacturer (MMM) framework, which enhances the resolution of both original and tampered images to improve comparison. Simultaneously, the framework converts the original and tampered images into feature embeddings and concatenates them, effectively modeling the context. Additionally, we used our MMM framework to create the Manipulation Mask Manufacturer Dataset (MMMD), which covers a wide range of manipulation techniques. We aim to contribute to the fields of image forensics and manipulation detection by providing more realistic manipulation data through MMM and MMMD. Detailed information about MMMD and the download link can be found at: https://github.com/ndyysheep/MMMD.

Keywords: Content security of visual media, image manipulation localization(IML), Datasets of visual media, Datasets Generation, Arbitrary-Scale Super-Resolution.

1. Introduction

Advances in digital image processing [5] have made software like Adobe Photoshop [42] and GIMP [18] more powerful, facilitating widespread image manipulation. Increasingly, there are examples of false information, retouched photographs, or edited video being released on social media. In many cases, this information goes ”viral” in just days, even hours [17]. This proliferation of false information and manipulated images threatens public knowledge, trust, and safety. Thus, image manipulation localization has emerged, and in some literature, it is also referred to as ”forgery detection [16]” or ”tamper detection [45].” Its purpose is to discern whether an input image is manipulated or authentic and to depict the exact manipulated parts of an image through a mask [40]. These parts are semantically different from the original content (the original image before manipulation). It does not include purely generated images (e.g., images generated from pure text) or the introduction of noise or other non-semantic changes through image processing techniques that do not alter the underlying meaning of the image. Standard tampered images and their masks are shown in Fig. 1.

However, a common benchmarking image dataset for algorithm evaluation and fair comparison is still lagging behind [10]. Most existing datasets are manually created and annotated by researchers [49], with limited tampering types and techniques, and the volume of datasets is also restricted. This leads to models with poor generalization and robustness. Therefore, we thought of creating datasets by sourcing a large number of original and manipulated images from the internet. But we found that images and videos from the

*Corresponding author.

Figure 1. Tampered images and their corresponding masks for the image manipulation localization task.

internet suffer from compression and clarity issues [4], and simply subtracting the original and manipulated images results in noisy images, as shown in the third row of Fig. 2, that traditional methods struggle to clean up.

Change detection [37, 30] involves identifying differences at the same location over different times, which is similar to our task of detecting differences between the original and manipulated images. Inspired by this, we treat the original and manipulated images as changes over time for the same picture, viewing the dataset generation task as a change detection task. However, due to the clarity disparity [41] between the two images in our task, directly using change detection models is not ideal. Therefore, we introduced a super-resolution processing module to enhance the details of the two images before generating the mask. This is the main idea behind our proposed Manipulation Mask Manufacturer (MMM) framework. Some of the masks we generated and their corresponding original images and tampered images are shown in Fig.2.

The framework pipeline inputs the original and tampered images into a Fully Convolutional Network (FCN) [28] to extract high-level features. These features are aligned using Maximum Mean Discrepancies (MMD) [38, 29] and concatenated. They are then processed through a Cross-scale Local Attention Block (CSLAB) [7] and a Local Frequency Encoding Block (LFEB) [7] to enhance resolution and detail. Finally, the features are split, decoded, subtracted, and a mask is obtained.

Our framework has achieved excellent annotation results on the IMD2020 [35], NIST16 [21], and CASIAv2 [13] datasets. Additionally, we created the Manipulation Mask Manufacturer Dataset (MMMD), containing 11,069 original images, tampered images, and masks, with potential for continuous growth. The dataset includes various resolutions and manipulation types, such as copy-move [1, 8], splicing [23], transformation [36], Deepfake [34], Image Inpainting [14], Image Morphing [47], Reconstruction [11], and Image Style Transfer [24]. It features diverse images like cartoons, portraits, landscapes, interiors, food, and accessories. The main parameters of our dataset compared to existing datasets are shown in Table 1. We used MMMD to train and test MVSS-Net [12] and IML-ViT [31]. Other 10 models pre-trained on CASIAv2 [13] struggled to achieve high metrics on our dataset, while models trained on our dataset demonstrated better generalization. This highlights the limitations of existing datasets.

In summary, our contributions are as follows:

- We propose a Manipulation Mask Manufacturer (MMM) framework that can accurately annotate the differences between original and tampered images even when there is a significant disparity in their clarity.
- We generated a large and diverse Manipulation Mask Manufacturer Dataset (MMMD) using the MMM framework to address the shortage of datasets in the field of image manipulation detection.
- Models pre-trained with our MMMD achieved higher F1 scores and demonstrated better generalization. Other pre-trained models struggled to perform well on our dataset, highlighting the limitations of existing tampering detection datasets. Our dataset better reflects real-world tampering scenarios.

2. RELATED WORK

2.1. Existing dataset generation methods

Current methods for generating image manipulation detection datasets include manual manipulation, which involves editing images by hand using tools like Adobe Photoshop [42], a time-consuming and expertise-demanding process [43]. Automatic manipulation employs software tools and scripts to rapidly produce large volumes of data, though these images may look unnatural or have obvious manipulation traces. Image synthesis combines elements from different images to create new visual scenes, enhancing dataset diversity but requiring complex techniques and substantial processing time [46]. The improved Total Variation Denoising Method [49] automatically subtracts a tampered image from the original to obtain a noisy mask, which

Figure 2. MMM framework generated result images. From highest to lowest, the sequence is as follows: original image, tampered image, image obtained by directly subtracting the two images and binarizing with a threshold of 30, and MMM predicted image.

Table 1. The primary datasets in the field of image manipulation localization. Most of these datasets suffer from issues such as limited quantity, single type of tampering, and inability to grow further, and they are all tampered with by the issuers of the datasets themselves.

Dataset	Tampered Images	Image Sources		Dataset Growth		Type	
		Self	Others	Scalable	Non-scalable	Traditional	AI-manipulated
Columbia	180	✓	–	–	✓	✓	–
CASIAv1	920	✓	–	–	✓	✓	–
CASIAv2	5,063	✓	–	–	✓	✓	–
Coverage	100	✓	–	–	✓	✓	–
NIST16	564	✓	–	–	✓	✓	–
DEFACTO	149,587	✓	–	–	✓	✓	–
IMD20	2,010	✓	–	–	✓	✓	✓
MMMD(Ours)	11,069	–	✓	✓	–	✓	✓

is then denoised, but this method often fails to convert many types of tampered images due to diverse noise and low generalization capability of traditional methods.

2.2. Existing change detection methods

In recent years, change detection methods based on deep learning have rapidly developed. Hao Chen et al. proposed the Bitemporal Image Transformer (BIT) [6], which converts input images into a small number of semantic tokens, uses a transformer encoder to model contextual information within the compact token space-time domain, and then projects the context-rich tokens back into the pixel space through a decoder to enhance the original features. The final change detection results are generated through feature difference images. ChangeFormer [3] uses a transformer-based Siamese network architecture for change detection from a pair of co-registered remote sensing images. This method combines a hierarchical transformer encoder and a multi-layer perceptron (MLP) decoder, effectively extracting multi-scale long-range feature differences. This is similar to image manipulation localization, and multi-scale techniques are also commonly used in decoders [51]. In SNUNet-CD [15], ChangeFormer extracts multi-scale features of bitemporal images through a hierarchical transformer encoder and generates change detection maps by fusing these feature differences using a lightweight MLP decoder. However, since the change detection considers the same image at different times with consistent clarity, it does not need to account for image degradation. Therefore, existing change detection models are not effective for our data generation tasks.

2.3. Existing tampered datasets

The development of datasets in the field of image manipulation detection has been relatively slow. Currently, widely recognized and used datasets are still those from four or five

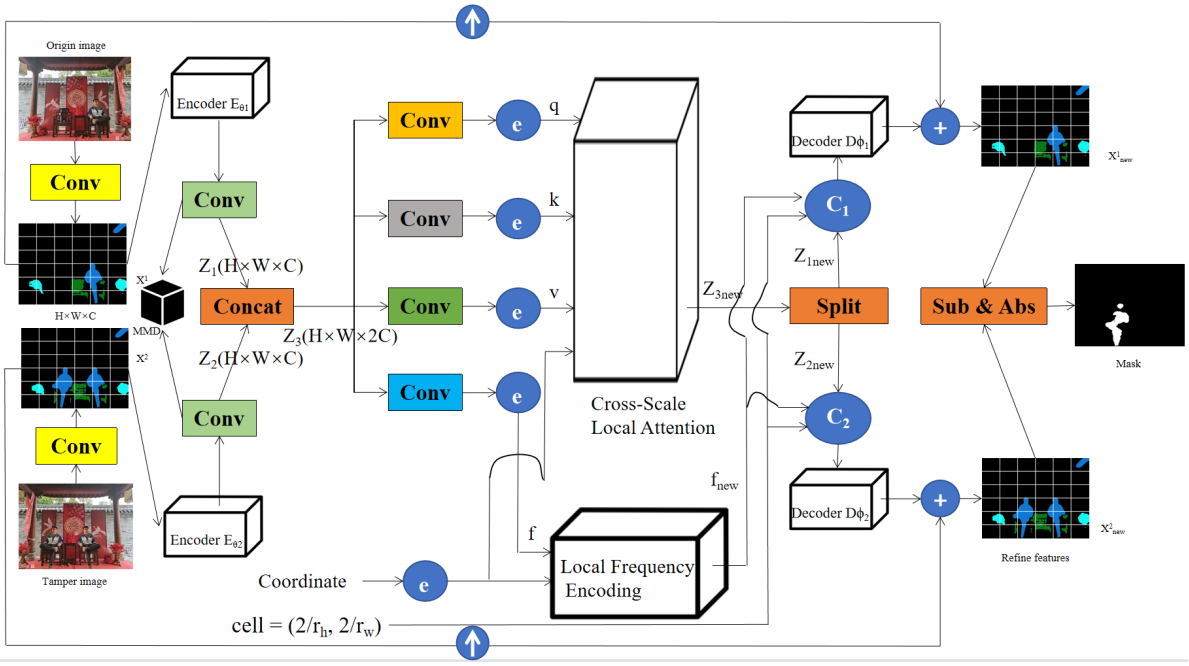

Figure 3. The proposed MMM framework. The local sampling operation samples input embeddings based on a grid of coordinates.

years ago, or even from over a decade ago.

Datasets for Traditional Tampering Techniques Almost all datasets include traditional tampering methods like splicing [23], copy-move [1, 8], removal, and various image enhancements to produce "fake" or "forged" images. Columbia [39] uses cropping and splicing [23], embedding parts from other images into a single image. CASIAv1 [13] employs Adobe Photoshop [42] for cutting and pasting, including geometric transformations [36] like scaling [2] and rotation [19]. CASIAv2 [13] adds more post-processing and has a richer variety of images, divided into eight categories: scenes, animals, buildings, people, plants, objects, nature, and textures. COVERAGE [46] consists of real images taken with an iPhone 6 front camera, processed with Photoshop CS4 using methods like translation, scaling, rotation, free transformation [36], lighting changes, and combinations thereof. NIST [21] uses local pixel modification, compression, noise addition, blurring, and geometric transformations [36]. DEFACTO [33], based on the MSCOCO [26] database, aims to produce semantically meaningful forged images, including splicing [23], copy-move [1, 8], object removal [9], and warping [20].

Deep Learning-Based Tampering Datasets Modern image tampering techniques have achieved unprecedented realism through artificial intelligence and deep learning, particularly with Generative Adversarial Networks (GANs). These techniques include deepfakes [34], which can perform facial replacement, expression synthesis, and generate images of non-existent people, making image and video tampering very realistic [52]. Deep learning also excels in image restoration and enhancement by denoising, filling in missing parts, and improving resolution, thus making damaged images look new and low-resolution images appear clear and detailed. Tools and frameworks such as TensorFlow, PyTorch, Keras, and OpenCV have greatly simplified the implementation and application of these techniques. In IMD2020 [35], GANs were used to generate tampered regions of images, and inpainting techniques [14] were employed to fill in missing or damaged parts of images, making them appear natural and coherent. This also presents greater challenges for image manipulation detection. Current models are increasingly in need of diverse data that better reflects real-world scenarios.

3. PROPOSED METHOD

3.1. The Pipeline of the Entire Framework

The entire MMM framework is divided into three modules: feature extraction and concatenation, super-resolution processing, and feature separation mask generation. We obtain the original images and a large number of tampered images from the network, keeping only the images of the same size as our original data. After obtaining the original and tampered images, we input them into our Manipulation Mask Manufacturer (MMM) framework. The MMM framework extracts high-level features from the original and tampered images using a Fully Convolutional Network (FCN) [28]. These features are aligned with Maximum Mean Discrepancies (MMD) [38, 29] and concatenated.

Table 2. Performance of Our Model on Different Datasets.

Dataset	F1	Precision	Recall	IoU	Accuracy
IMD2020	0.88	0.90	0.87	0.81	0.97
NIST16	0.94	0.95	0.92	0.89	0.98
CASIAv2	0.95	0.96	0.95	0.91	0.98

The idea of concatenating high-level features is inspired by the Bitemporal Image Transformer (BIT) [6]. During super-resolution, the concatenated features are processed by the Cross-scale Local Attention Block (CSLAB) [7] and the Local Frequency Encoding Block (LFEB) [7] to enhance the resolution and detail representation of the images. The framework then separates these embeddings, uses decoders to generate residual images, and combines them with the original images. The final mask is produced by computing the absolute difference between the high-resolution features of the original and tampered images. The entire MMM structure is shown in Fig.3.

3.2. Specific Processing Algorithm

Extraction and Concatenation of Image Features First, we use a Fully Convolutional Network (FCN) [28] to extract high-level features from the original image and the tampered image, respectively. Then, these features are input into encoder $E_{\theta 1}$ and encoder $E_{\theta 2}$, resulting in the feature embeddings $Z_1 \in \mathbb{R}^{H\times W\times C}$ and $Z_2 \in \mathbb{R}^{H\times W\times C}$. Z_1 and Z_2 are subjected to Maximum Mean Discrepancies (MMD) [38, 29] calculation to eliminate the differences in data distribution, allowing the model to focus more on the differences in content. Simultaneously, Z_1 and Z_2 are concatenated into $Z_3 \in \mathbb{R}^{H\times W\times 2C}$.

Arbitrary-Scale Super-Resolution Z_3 will be projected by four separate convolutional layers to obtain four latent embeddings, corresponding to query q, key k, value v, and frequency f. Since the sizes of the two images are the same, we use the coordinates and cell of the original image. The original image and the tampered image will generate 2D high-resolution coordinates based on an arbitrary upsampling scale $\mathbf{r} = \{r_h, r_w\}$ in 2D low-resolution coordinates. Next, the 2D coordinates, along with q, k, and v, will be input into the Cross-Scale Local Attention Block (CSLAB) [7] to estimate a local latent embedding $Z_{3new} \in \mathbb{R}^{G_h G_w\times 2C}$. f and the 2D coordinates will also be input into the Local Frequency Encoding Block (LFEB) [7] to estimate a local frequency embedding $f_{new} \in \mathbb{R}^{G_h G_w\times 2C}$. Specifically, G_h and G_w represent the height and width of the local grids used for performing local coordinate sampling. CSLAB [7] and LFEB [7] estimate Z_{3new} and f_{new} as follows:

$$Z_{3new} = CSLAB(\delta x, q, k, v) \tag{1}$$

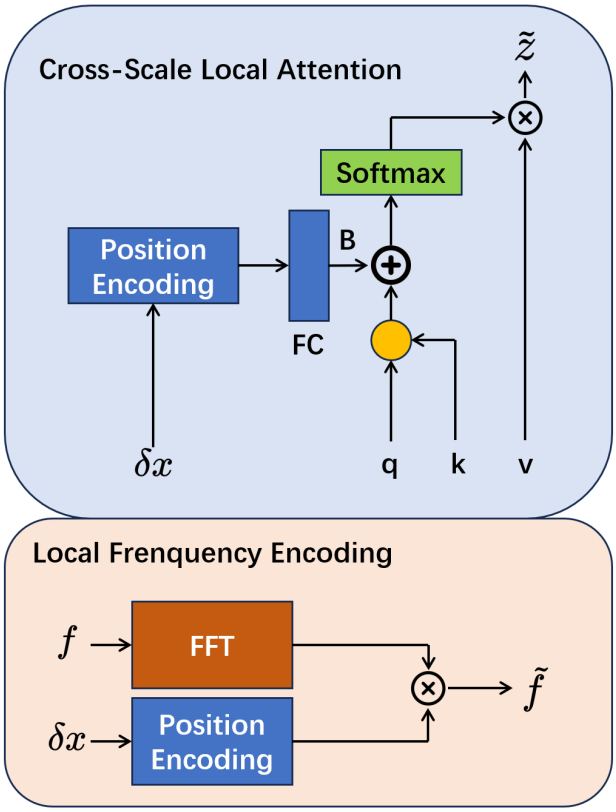

Figure 4. The framework of CSLAB and LFEB.

$$f_{new} = \text{LFEB}(\delta x, f) \tag{2}$$

$$\delta \mathbf{x} = \left\{ x_q - x^{(i,j)} \right\}_{i\in\{1,2,\ldots,G_h\}, j\in\{1,2,\ldots,G_w\}} \tag{3}$$

CSLAB and LFEB draw on the work of Chen et al. [7], with specific structures shown in Fig. 4. The primary function of the Cross-scale Local Attention Block (CSLAB) is to aggregate cross-scale local feature information, utilizing query, key, and value features to enhance the model's ability to capture fine-grained details, thereby improving the accuracy of image manipulation detection. The Local Frequency Encoding Block (LFEB) is responsible for performing local frequency encoding on the input frequency features, enhancing the model's sensitivity to edge and texture information by capturing local frequency variations.

Separate image features and generate a mask Z_{1new} and Z_{2new} are derived from splitting Z_{3new}. Decoder $\mathrm{D}\phi_1$

Table 3. The F1 scores of various models pre-trained on MMMD across different datasets. Left: Pre-trained with CASIAv2, Right: Pre-trained with MMMD (ours).

Model \ Dataset	COVERAGE	Columbia	NIST16	IMD2020
MVSS-Net	0.26/**0.34**	0.39/**0.49**	0.25/**0.28**	0.28/**0.32**
IML-ViT	0.43/0.29	0.78/**0.81**	0.33/**0.34**	0.33/**0.37**

and Decoder $\mathrm{D}\phi_2$ respectively utilize these embeddings along with the provided cell to generate residual images. Then, the original and tampered images are upsampled and added element-wise to their corresponding residual images. This results in the high-resolution high-level features of the original and tampered images. We subtract these two high-resolution high-level features and take the absolute value to obtain the final mask.

Loss Function The loss function of the model is defined as follows:

$$L = \frac{1}{H_p \times W_p} \sum_{h=1}^{H} \sum_{w=1}^{W} l\left(P_{hw}, M_{hw}\right) \tag{4}$$

Here, $H_p \times W_p$ is the total number of pixels in the image, P_{hw} is the probability distribution of the model at (h, w), and M_{hw} is the mask label at position (h, w). $l(P_{hw}, M_{hw})$ is the cross-entropy, which is calculated as:

$$l(P_{hw}, M_{hw}) = -\sum_{c} M_{hw}^{(c)} \log(P_{hw}^{(c)}) \tag{5}$$

where c indexes the classes, $M_{hw}^{(c)}$ is a binary indicator (0 or 1) if class label c is the correct classification for position (h, w), and $P_{hw}^{(c)}$ is the predicted probability of class c at position (h, w). The cross-entropy measures the dissimilarity between the true label distribution and the predicted probability distribution, and it is used to optimize the model parameters by minimizing this dissimilarity.

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1. Creation of Manipulation Mask Manufacturer Dataset (MMMD)

We crawled images from websites containing tampered images on the internet, such as Baidu PS Bar. It is an ideal source for both original and tampered images [49]. Most users request others to help modify the pictures they provide. As a result, there are often numerous tampered images under their posts. We save the original and tampered images from different posts, allowing us to collect a large amount of data in a short period. We take the first image of each post as the original image, like the first row in Fig. 2, and consider all other images of the same size in that post as tampered images, like the second row in Fig. 2.

We then subtract the grayscale images of the original and tampered images to obtain a mask image containing a significant amount of noise. This is because images undergo irreversible compression and quality degradation during transmission over the internet, and they also experience compression when opened with image editing software like Photoshop. The original and tampered images contain noise differences that are imperceptible to the human eye, making the directly subtracted mask image unsuitable for use as training data. The images in the NIST16 dataset are multi-scale, so we used it to pre-train the MMM framework. Therefore, we input the original and corresponding tampered images as pairs into our MMM framework, which consists of three steps: feature extraction and concatenation, super-resolution processing, and feature separation mask generation. This process ultimately produces the predicted mask image shown in the last row of Fig. 2. The predicted image has significantly reduced noise and can be used for tamper detection models. Since the first image from the PS forum is assumed to be original and subsequent ones tampered, but this isn't always true, it leads to noisy, mostly white masks. Special tampering techniques also cause noisy masks. Therefore, masks with more than 70% or less than 1% white area are deemed invalid and removed.

The entire MMMD is divided into three groups: original images, tampered images, and predicted masks, each containing 11,069 images. Each tampered image has a corresponding original image and mask in the other two groups. The dataset contains images with different resolutions and various manipulation types, including copy-move[1, 8], transformation [36], Deepfake [34], image inpainting [14], morphing [47], reconstruction [11], and style transfer. It encompasses a wide range of image categories, such as cartoons, portraits, landscapes, interiors, food, and accessories.

4.2. Accuracy of the Model on Existing Datasets

Our innovative use of deep learning for annotating image manipulation detection datasets has no existing comparable methods. Thus, we train and validate our model on the NIST16 [21], IMD2020 [35], and CASIAV2.0 [13] datasets to demonstrate its effectiveness in distinguishing between original and tampered images. The experimental results are shown in Table 2. The model performs exceptionally

Table 4. The F1 scores of various models pre-trained on CASIAv2 across different datasets.

Model \ Dataset	COVERAGE	Columbia	CASIAv1	MMMD(Ours)
Mantra-Net	0.08	0.46	0.12	**0.09**
MVSS-Net	0.26	0.39	0.53	**0.26**
CAT-Net	0.30	0.58	0.58	**0.30**
ObjectFormer	0.29	0.34	0.43	**0.32**
NCL-IML	0.22	0.45	0.50	**0.26**
TruFor	0.42	0.86	0.72	**0.30**
IML-ViT	0.43	0.78	0.72	**0.24**
PSCC-Net	0.23	0.60	0.38	**0.32**

well on all three datasets, with high levels across all metrics (F1, Precision, Recall, IoU, and Accuracy). It demonstrates strong generalization capability on image manipulation detection datasets, particularly on datasets involving traditional tampering methods. The performance is slightly lower on the IMD2020 [35], which uses deep learning techniques such as GANs and inpainting [14], but overall, the model exhibits good adaptability and performance across various datasets.

Implementation Details Our models are implemented on PyTorch. Our training is set with a learning rate of 0.01 and a maximum of 100 epochs. The learning rate decay iterations are set to 100. Validation is performed after each training epoch, and the model that performs best on the validation set is used for evaluation on the test set.

Evaluation Metrics We use the F1-score to evaluate the performance of our model. It balances between tamper detection (recall) and avoiding false positives (precision), preventing the bias that comes from solely pursuing high recall or high precision. The formula for F1-score is as follows:

$$\text{F1-score} = 2 \times \left(\frac{\text{Precision} \times \text{Receall}}{\text{Precision} + \text{Recall}} \right) \tag{6}$$

Precision and Recall are expressed using the four metrics: True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). Additionally, we use IoU (Intersection over Union) to represent the model's localization accuracy and Accuracy to evaluate the overall performance of the model. The formula of IoU is $\text{IoU} = \text{TP}/(\text{TP} + \text{FN} + \text{FP})$ and The formula of Accuracy is $\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FN} + \text{FP})$.

4.3. Effect of the Generated Dataset on Existing Models

IMDL-BenCo [32] reproduces mainstream IML models, and our model experiments are based on this framework.

We used our MMMD to train two major models, MVSS-Net [12], and IML-ViT [31], and validated their generalization. The results are shown in Table 3.

As shown in the table, MVSS-Net and IML-ViT trained using MMMD achieved higher metrics across various datasets compared to models pre-trained on CASIAv2. They exhibit higher generalization and robustness on MMMD. Since MMMD is much larger than commonly used datasets and covers a wider variety of manipulation types and scenarios, it helps improve the generalization ability of existing manipulation detection models. Models pre-trained on MMMD are expected to achieve better performance on other datasets, which is consistent with our experimental results.

We used MMMD to validate image manipulation detection models Mantra-Net [48], MVSS-Net [12], CAT-Net [25], ObjectFormer [44], NCL-IML [50], TruFor [22], IML-ViT [31] and PSCC-Net [27], pre-trained on CASIAv2 [13] and discovered the shortcomings of existing datasets compared to our MMMD. The results are shown in Table 4.

As shown in the table, various models pre-trained on CASIAv2 [13] struggled to achieve high metrics on MMMD. CASIAv2 is one of the larger datasets in recent years for manipulation detection, covering a wide range of manipulation types and commonly used as a pre-training dataset for models. However, models pre-trained on CASIAv2 exhibited lower metrics on our MMMD compared to other datasets, highlighting the limitations of traditional datasets. In contrast to our MMMD, these datasets are smaller in size, cover fewer manipulation types, and are still somewhat distant from real-world manipulated images. As a result, models trained on these datasets struggle to perform well on real-life manipulated images.

Evaluation Metrics We also use the F1-score to measure the accuracy of the model's detection, with the calculation formula given in Equation 6.

Effect Without Using Super-Resolution When processing images directly without using the super-resolution module, for certain severely degraded images, we still ob-

tain noisy images without super-resolution. These images are not suitable for use as training data for manipulation detection models.

5. CONCLUSION

In this paper, we creatively propose a Manipulation Mask Manufacturer (MMM) framework for generating image manipulation detection datasets. It addresses the issues of small dataset size, poor quality, and limited types of tampering detection in the field of image manipulation detection. It concatenates image feature embeddings, performs context modeling, and captures long-range relationships between pixels. It uses MMD to eliminate the differences in data distribution. Extensive experiments have validated the effectiveness of our method. We demonstrated the strong performance of the MMM framework on existing datasets. The MMMD dataset we proposed better aligns with the real-world tampering scenarios that manipulation detection models must face. This will help these models improve their generalization and robustness in practical applications. We also demonstrated that MMMD outperforms other datasets in training effectiveness.

References

[1] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra. A sift-based forensic method for copy–move attack detection and transformation recovery. *IEEE transactions on information forensics and security*, 6(3):1099–1110, 2011. 2, 4, 6

[2] I. Andreadis and A. Amanatiadis. Digital image scaling. In *2005 IEEE Instrumentationand Measurement Technology Conference Proceedings*, volume 3, pages 2028–2032. IEEE, 2005. 4

[3] W. G. C. Bandara and V. M. Patel. A transformer-based siamese network for change detection. In *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 207–210, 2022. 3

[4] T. Boudier and D. M. Shotton. Video on the internet: An introduction to the digital encoding, compression, and transmission of moving image data. *Journal of structural biology*, 125(2-3):133–155, 1999. 2

[5] K. R. Castleman. *Digital image processing*. Prentice Hall Press, 1996. 1

[6] H. Chen, Z. Qi, and Z. Shi. Remote sensing image change detection with transformers. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2021. 3, 5

[7] H.-W. Chen, Y.-S. Xu, M.-F. Hong, Y.-M. Tsai, H.-K. Kuo, and C.-Y. Lee. Cascaded local implicit transformer for arbitrary-scale super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18257–18267, 2023. 2, 5

[8] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou. An evaluation of popular copy-move forgery detection approaches. *IEEE Transactions on information forensics and security*, 7(6):1841–1854, 2012. 2, 4, 6

[9] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004. 4

[10] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato. Raise: A raw images dataset for digital image forensics. In *Proceedings of the 6th ACM multimedia systems conference*, pages 219–224, 2015. 1

[11] G. Demoment. Image reconstruction and restoration: Overview of common estimation structures and problems. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(12):2024–2036, 1989. 2, 6

[12] C. Dong, X. Chen, R. Hu, J. Cao, and X. Li. Mvss-net: Multi-view multi-scale supervised networks for image manipulation detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3539–3553, 2022. 2, 7

[13] 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*, pages 422–426. IEEE, 2013. 2, 4, 6, 7

[14] O. Elharrouss, N. Almaadeed, S. Al-Maadeed, and Y. Akbari. Image inpainting: A review. *Neural Processing Letters*, 51:2007–2028, 2020. 2, 4, 6, 7

[15] S. Fang, K. Li, J. Shao, and Z. Li. Snunet-cd: A densely connected siamese network for change detection of vhr images. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022. 3

[16] H. Farid. Image forgery detection. *IEEE Signal processing magazine*, 26(2):16–25, 2009. 1

[17] N. Fitzpatrick. Media manipulation 2.0: the impact of social media on news, competition, and accuracy. 2018. 1

[18] G. GIMP. Gimp. *Página Principal. Disponível em*, 6, 2013. 1

[19] R. H. Ginsberg. Image rotation. *Applied optics*, 33(34):8105_1–8108, 1994. 4

[20] C. A. Glasbey and K. V. Mardia. A review of image-warping methods. *Journal of applied statistics*, 25(2):155–171, 1998. 4

[21] H. Guan, M. Kozak, E. Robertson, Y. Lee, A. N. Yates, A. Delgado, D. Zhou, T. Kheyrkhah, J. Smith, and J. Fiscus. Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 63–72. IEEE, 2019. 2, 4, 6

[22] F. Guillaro, D. Cozzolino, A. Sud, N. Dufour, and L. Verdoliva. Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20606–20615, 2023. 7

[23] Y.-F. Hsu and S.-F. Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *2006 IEEE International Conference on Multimedia and Expo*, pages 549–552. IEEE, 2006. 2, 4

[24] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, 26(11):3365–3385, 2019. 2

[25] M.-J. Kwon, S.-H. Nam, I.-J. Yu, H.-K. Lee, and C. Kim. Learning jpeg compression artifacts for image manipulation detection and localization. *International Journal of Computer Vision*, 130(8):1875–1895, 2022. 7

[26] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 4

[27] X. Liu, Y. Liu, J. Chen, and X. Liu. Pscc-net: Progressive spatio-channel correlation network for image manipulation detection and localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(11):7505–7517, 2022. 7

[28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2, 4, 5

[29] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015. 2, 4, 5

[30] D. Lu, P. Mausel, E. Brondizio, and E. Moran. Change detection techniques. *International journal of remote sensing*, 25(12):2365–2401, 2004. 2

[31] X. Ma, B. Du, X. Liu, A. Y. A. Hammadi, and J. Zhou. Iml-vit: Image manipulation localization by vision transformer. *arXiv preprint arXiv:2307.14863*, 2023. 2, 7

[32] X. Ma, X. Zhu, L. Su, B. Du, Z. Jiang, B. Tong, Z. Lei, X. Yang, C.-M. Pun, J. Lv, et al. Imdl-benco: A comprehensive benchmark and codebase for image manipulation detection & localization. *arXiv preprint arXiv:2406.10580*, 2024. 7

[33] G. Mahfoudi, B. Tajini, F. Retraint, F. Morain-Nicolier, J. L. Dugelay, and P. Marc. Defacto: Image and face manipulation dataset. In *2019 27Th european signal processing conference (EUSIPCO)*, pages 1–5. IEEE, 2019. 4

[34] Y. Mirsky and W. Lee. The creation and detection of deepfakes: A survey. *ACM computing surveys (CSUR)*, 54(1):1–41, 2021. 2, 4, 6

[35] A. Novozamsky, B. Mahdian, and S. Saic. Imd2020: A large-scale annotated dataset tailored for detecting manipulated images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 71–80, 2020. 2, 4, 6, 7

[36] M. M. Petrou and C. Petrou. *Image processing: the fundamentals*. John Wiley & Sons, 2010. 2, 4, 6

[37] R. A. Rensink. Change detection. *Annual review of psychology*, 53(1):245–277, 2002. 2

[38] D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The annals of statistics*, pages 2263–2291, 2013. 2, 4, 5

[39] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. 4

[40] L. Su, X. Ma, X. Zhu, C. Niu, Z. Lei, and J.-Z. Zhou. Can we get rid of handcrafted feature extractors? sparsevit: Nonsemantics-centered, parameter-efficient image manipulation localization through spare-coding transformer. *arXiv preprint arXiv:2412.14598*, 2024. 1

[41] Y. Tan, H. Zheng, Y. Zhu, X. Yuan, X. Lin, D. Brady, and L. Fang. Crossnet++: Cross-scale large-parallax warping for reference-based super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4291–4305, 2020. 2

[42] A. C. Team and B. Gyncild. *Adobe Photoshop CC*. Pearson Education, 2013. 1, 2, 4

[43] R. Thakur and R. Rohilla. Recent advances in digital image manipulation detection techniques: A brief review. *Forensic science international*, 312:110311, 2020. 2

[44] J. Wang, Z. Wu, J. Chen, X. Han, A. Shrivastava, S.-N. Lim, and Y.-G. Jiang. Objectformer for image manipulation detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2364–2373, 2022. 7

[45] W. Wang, J. Dong, and T. Tan. A survey of passive image tampering detection. In A. T. S. Ho, Y. Q. Shi, H. J. Kim, and M. Barni, editors, *Digital Watermarking*, pages 308–322, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. 1

[46] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler. Coverage—a novel database for copy-move forgery detection. In *2016 IEEE international conference on image processing (ICIP)*, pages 161–165. IEEE, 2016. 2, 4

[47] G. Wolberg. Image morphing: a survey. *The visual computer*, 14(8-9):360–372, 1998. 2, 6

[48] Y. Wu, W. AbdAlmageed, and P. Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9543–9552, 2019. 7

[49] X. Yang and J. Zhou. Manipulation mask generator: High-quality image manipulation mask generation method based on modified total variation noise reduction. In *2023 IEEE 4th International Conference on Pattern Recognition and Machine Learning (PRML)*, pages 218–223. IEEE, 2023. 1, 2, 6

[50] J. Zhou, X. Ma, X. Du, A. Y. Alhammadi, and W. Feng. Pre-training-free image manipulation localization through non-mutually exclusive contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22346–22356, 2023. 7

[51] X. Zhu, X. Ma, L. Su, Z. Jiang, B. Du, X. Wang, Z. Lei, W. Feng, C.-M. Pun, and J. Zhou. Mesoscopic insights: Orchestrating multi-scale & hybrid architecture for image manipulation localization. *arXiv preprint arXiv:2412.13753*, 2024. 3

[52] P. Zhuang, H. Li, S. Tan, B. Li, and J. Huang. Image tampering localization using a dense fully convolutional network. *IEEE Transactions on Information Forensics and Security*, 16:2986–2999, 2021. 4