# FUSION++: A Method to Detect Generative AI Manipulated Images

Lamyaa Aljuaid
L.Z.M.Aljuaid2@newcastle.ac.uk
Deepayan Bhowmik
deepayan.bhowmik@newcastle.ac.uk
Department of Computing
Newcastle University
Newcastle, UK

**Abstract**

Modern-day image manipulation techniques, particularly AI-powered ones, are a significant concern today as they can easily fool the human eyes, leading to the misuse, forgery, and propagation of disinformation by creating deceptive visual content. There is a need to accurately detect manipulation and localization of the manipulated region to identify where manipulation happened in the image. This paper introduces NCL_IMD.v2, a new dataset for AI-manipulated images and a manipulation detection technique, FUSION++, which constitutes progress on state-of-the-art detection algorithms such as MMFUSION. Unique in its kind, the proposed new dataset provides an orderly manipulation technique that uses generative AI and prompts engineering to produce real-like outcomes. The detection algorithm, FUSION++, used an additional HOG-based feature extractor alongside other feature extractors used in MMFUSION and also incorporated a shifted window-based attention mechanism. The results of these integrations showed major performance improvements. FUSION++ consistently produced higher detection and localization performance on both existing datasets, such as AutoSlice and CocoGlide, and the newly introduced dataset, providing a robust solution for AI-manipulated visual content.

## 1 Introduction

Image manipulation is an emerging topic of interest, especially due to the emergence of generative AI that can produce real-like images in an automated fashion. Due to the availability of sophisticated visual tools and research breakthroughs in Artificial Intelligence (AI), it is difficult or impossible to identify manipulated images with the naked eye, and even software cannot detect real manipulated images. Manipulated images can be of several types: (1) part of the image modified by adding, removing, or altering the objects, and (2) generating completely new, real-like images. Manipulated images are misused in politics, document forgery, and fake opinion creation. As per human psychology, manipulated visual content can easily fool humans [21]. There is a severe need to develop methods to detect such manipulated images before it harms an individual, group, or organization.

The techniques used to manipulate images can be divided into two types: (i) manual and (ii) AI-based techniques. Manual techniques refer to manipulating authentic images using conventional methods such as Copy-Move, and Splicing. In Copy-Move [8], modifications

are performed within the same image. It involves copying an object or portion from the image and pasting it into another location within the same image typically used to hide information. In Splicing techniques [30], a manipulated image is a composite created by combining two or more images.AI-based techniques are another specific manipulation type that mainly focuses on altering an image using artificial intelligence approaches, particularly Generative Adversarial Networks (GANs) [8], and Diffusion Models (DMs) [20]. However, there is a handful of useful datasets currently available that are created using generative AI-based image manipulation.

In addressing such a gap, this work introduces NCL_IMD.v2[1] a new dataset that uses generative AI and prompt engineering-based image manipulation in an automated way. Our finding suggests existing state-of-the-art detection algorithms perform poorly on the new dataset, and therefore, we also propose a new detection technique by progressing state-of-the-art with an aim to primarily detect AI-manipulated images. The contributions of the paper are the following:

- Development of a well-organized AI-based image manipulation dataset and
- Introduction of a new detection algorithm for generative AI-manipulated images, and it has significantly outperformed state-of-the-art techniques on the new dataset.

# 2 Related Work

## 2.1 Datasets

Datasets play a crucial role in evaluating the generalizability and robustness of image manipulation detection methods; the current datasets can be summarized into two categories: manual manipulation and generative AI. Figure 1 illustrates the organization of available datasets. Most of the existing datasets belong to the first category, in which images are manipulated manually using Copy-Move and Splicing. The most popular copy-move datasets are COVERAGE [27], CoMoFoD [24], MICC F220/F2000 [1], FAU/Manip[6], GRIP [7] ,and the common splicing datasets are Columbia [10], Carvalho [4], Wild Web [29],VIPP Real [2] and MISD [12].

In contrast, there is a limited number of datasets within the generative AI category, where the digital images are manipulated using advanced artificial intelligence generators such as generative adversarial networks and diffusion models. In [11], the AutoSplice dataset composed of 2273 original images and 3621 manipulated images, was generated using DALL-E2, where the editing process is guided by text prompts. Also, in the CocoGlide dataset [9], the 512 original images were manipulated by a Glide diffusion-based model. Both AutoSplice and CocoGlide focused on applying one manipulation type by one generator, which is replacing the masked regions based on the provided textual prompts. Thus, these datasets have limitations in terms of diversity, which may affect the generalizability of detection models trained on them. Based on the abovementioned literature, there is a lack of datasets that consist of diverse and AI-based manipulated images. This work proposed NCL_IMD.v2 a new manipulated image dataset containing comprehensive and sophisticated AI-based image manipulation.

[1] The NCL_IMD.v2 dataset is available at: https://dbhowmik.github.io/MediaTrust/dataset/

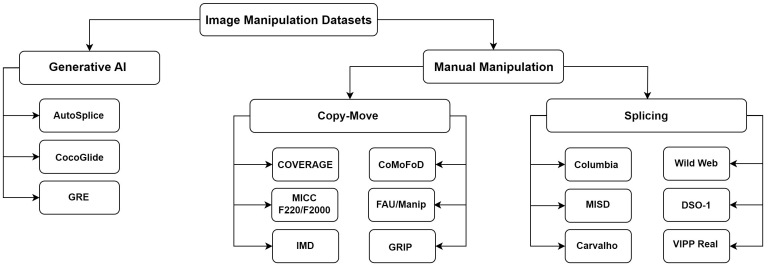


Figure 1: Classification of the currently available datasets into two distinct categories: Manual Manipulation and Generative AI.

## 2.2 Manipulation Detection

The current state-of-the-art models are designed to perform two tasks: detection to determine whether the input image is genuine or manipulated and localization to identify the manipulated regions within the image. Table 1 provides a summary of the state-of-the-art detection methods. To detect and localize image manipulations, Kwon et al. [14], proposed CAT-Net a convolutional neural network-based model that integrates RGB and DCT streams to analyze visual and compression artifacts. The PSCC-Net proposed in [17], consists of two paths Top-Down and Bottom-Up. In the first path, HRNetV2p-W18 was used to extract the local and global features, while the second path produced the detection score and localization map. MVSS-Net in [5], contains two branches: the Edge-Supervised branch and the Noise-Sensitive branch. These branches extract features from RGB images and Noise, then the Dual Attention mechanism was used to fuse the features from the two branches. The TruFor method proposed in [9], a SegFormer transformer-based architecture used to extract features from RGB image and Noiseprint++, then the Cross-Modal Feature Rectification module was used to combine these features and enhance the detection of anomalies. In MMFUSION [25], they extend the TruFor architecture by incorporating NoisePrint++, SRM, and Bayar convolution forensic filters and exploring different fusion techniques.

The review of the literature discussed above emphasizes the increasing importance and complexity of image manipulation technology in today's world. The research encompasses a range of topics, including the development of innovative detection algorithms and the uti-

Table 1: Summary of state-of-the-art image manipulation detection & localization methods.

| Method | Year | Architecture | | | Dataset source | |
|---|---|---|---|---|---|---|
| | | Features | Fusion | Detection and localization | Created | Used |
| **CAT-Net [14]** | 2021 | RGB and DCT | Early | Segmentation Network. | | ✓ |
| **PSCC-Net [17]** | 2021 | Local and Global. | Early | HRNetV2. Progressive mechanism and Spatio Channel Correlation(SCCM). | | ✓ |
| **MVSS-Net [5]** | 2021 | RGB and Noise. | Late | Global Max Pooling. Pixel-Wise Segmentation Map. | | ✓ |
| **TruFor [9]** | 2022 | RGB and Noisprint++ | Early | Anomaly localization map, Confidence map. Integrity Score. | ✓ | |
| **MMFUSION [25]** | 2023 | RGB, Noisprint++, SRM, and Bayar convolution. | Early & Late | Anomaly localization map, Confidence map. Detection score. | | ✓ |

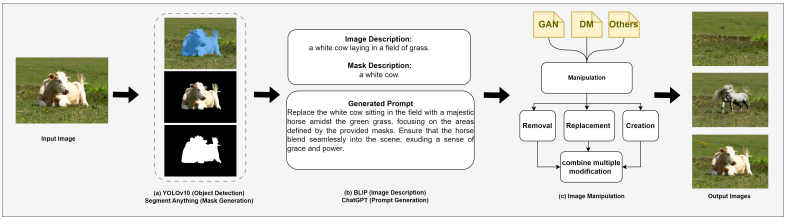


Figure 2: Illustration of the pipeline for NCL_IMD.v2 generation. (a): Object detection & mask generation, (b): Visual content analysis & prompt generation, and (c): Applying manipulation techniques to the image.

lization of methods involving various CNN, GANs, etc. Together, these studies demonstrate progress in countering the impacts of manipulated images. Ultimately, it represents an advancement in protecting media integrity and fostering public trust.

# 3 Methodology

## 3.1 Dataset Generation

Existing datasets for AI-manipulated image detection, such as AutoSplice [10], CocoGlide [9] and GRE [22] are limited in terms of diversity and availability. To address these issues, we developed a comprehensive dataset to advance the development of AI-manipulated image detection systems. NCL_IMD.v2 covers several manipulation types including removal, creation, replacement, and combination of manipulations using various AI generators. An automated methodology was used to generate NCL_IMD.v2 dataset. Figure 2 illustrates an overview of the process of creating the new dataset.

The source of the original images in our dataset is a subset with a size of 25K images from the COCO dataset [16]. To generate a high-quality mask with precise boundaries, we first detected the object using YOLOv10 [26], and then used the Segment Anything Model [13] to generate the mask. In some modification scenarios, providing a description of both the image and masked object plays a significant role in guiding the manipulation process, ensuring consistency and visual coherence, and the realism of the manipulated image. To generate the descriptions, we used BLIP [15], and ChatGPT [18] for prompt generation. In our dataset, we leverage a mix of generators, such LaMa [23], DALL-E [19], PowerPaint [31], and Paint by Example [28]. Figure 3 illustrates sample images from NCL_IMD.v2 dataset. Currently, the dataset consists of 12K original and manipulated images along with its ground truth and prompts. 12K images are selected randomly from the original set of 25K images to ensure a diverse and unbiased selection.

## 3.2 Detection Method

AI-manipulated image detection is an exciting area, particularly in today's age and from a future perspective. It has a wide application area. The problem with the current methods for manipulated image detection is that they are computationally exhaustive and incapable of handling AI-based image manipulations. One state-of-the-art method used in detecting manipulated images and localizing image manipulations is MMFUSION [25], achieved by fusing multiple forensic modalities. It has two significant steps: (1) feature extraction and

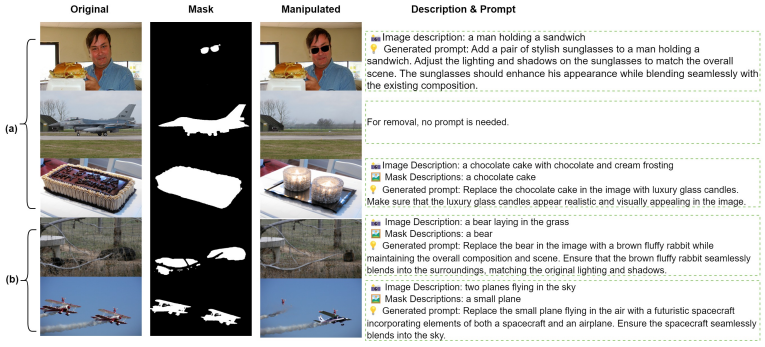


Figure 3: Sample from NCL_IMD.v2 dataset. (a): Single and (b): Multiple manipulations.

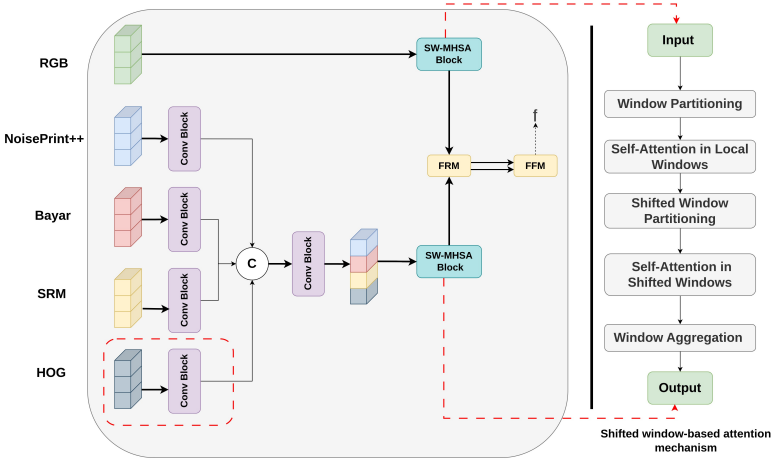


Figure 4: Overall architecture of the proposed FUSION++ manipulation detection technique.

(2) fusion. Firstly, it uses NoisePrint++, SRM, and Bayar Convolution filters for feature extraction, which can handle diverse manipulations. Secondly, early or late fusion techniques are used to learn the pattern from the extracted features for detection and localization.

This paper proposes an image manipulation detection and localization method known as FUSION++[2], which incorporates a Histogram of Oriented Gradients (HOG) for understanding minute manipulations by improving the quality of extracted features. HOG features integrated with FUSION++ increase its ability to capture the contextual relevance within manipulated images to detect subtle manipulations. HOG helps detect edges and gradients, making them ideal for manipulated image detection. It also complements NoisePrint++ which extracts artifacts related to camera and editing history, SRM which reveals noise and inconsistencies, and Bayar Convolution which extracts noise related to manipulation traces by detailing the spatial distribution of gradients. Further, FUSION++ uses a shifted window-based attention mechanism. On the other hand, the shifted window-based attention mechanism addresses two major drawbacks of MMFUSION's self-attention mechanism, which are (1) computational efficiency and (2) critical local context sensitivity. Further, Figure 4

[2]Code is available at: https://github.com/Lamyaa2050/FUSIONpp

depicts the high-level block diagram of the FUSION++. The high-level mathematical formulation of FUSION++ is given below:

**Feature Extraction**: These are the features used by MMFUSION [25]. The input image $\mathbf{I}$ is processed through Conv Blocks to generate different feature representations:

$$\mathbf{F}_{\text{RGB}}, \mathbf{F}_{\text{NP}}, \mathbf{F}_{\text{SRM}}, \mathbf{F}_{\text{BC}} = \text{Extract}(\mathbf{I}), \tag{1}$$

where features from RGB channels, NoisePrint++, Spatial Rich Model, and Bayar Convolution are denoted by $\mathbf{F}_{\text{RGB}}$, $\mathbf{F}_{\text{NP}}$, $\mathbf{F}_{\text{SRM}}$, and $\mathbf{F}_{\text{BC}}$.

**Early Fusion**: In FUSION++, an early fusion approach has been selected. This selection is purely based on the image manipulation problem. Early fusion helps in understanding the interactions between the various forensic modalities, which is essential when finding clues for manipulations. Using Early fusion, diverse features are combined at the initial stage, which helps FUSION++ learn complex relationships.

For Early Fusion, the extracted features are fused through a:

$$\mathbf{F}_{\text{combined}} = [\mathbf{F}_{\text{RGB}}, \mathbf{F}_{\text{NP}}, \mathbf{F}_{\text{SRM}}, \mathbf{F}_{\text{BC}}], \tag{2}$$

for processing and integration, the $\mathbf{F}_{\text{combined}}$ is fed to another Conv Block and CMX encoder.

**Detection and Localization**: The detector uses the fused feature map to identify manipulations and locate them in the image:

$$S, M = \text{Detector}(\mathbf{F}_{\text{combined}}), \tag{3}$$

where detection score, and localization map are denoted by $S$, and $M$.

**Incorporating HOG Features**: HOG add more context to above computed features which are used in MMFUSION. As depicted in Figure 5, visualizing HOG vectors reveals distinct patterns that differentiate authentic and AI-manipulated images. It is noticed that in manipulated regions, the intensity of HOG vectors tends to be significantly lower compared to the genuine parts of the image. Also, the directionality of HOG vectors in manipulated regions is more uniform and aligned, with vectors pointing predominantly in the same direction, which is vice versa in the case of authentic regions displaying HOG vectors with a wider range of directions, reflecting the natural variability in texture and structure. Thus, adding HOG in FUSION++ produces better detection and localization of manipulated areas within images. There are two major issues if standard HOG is used which are: (1) HOG image is a single-channel, making it incompatible with the model, and (2) its representation can lack continuity across the image. To handle this, in FUSION++, HOG feature extractors are used with different tile sizes (5x5, 7x7, and 9x9):

$$\mathbf{F}_{\text{HOG-5x5}}, \mathbf{F}_{\text{HOG-7x7}}, \mathbf{F}_{\text{HOG-9x9}} = \text{HOG}_{3\text{x}3}(\mathbf{I}), \text{HOG}_{6\text{x}6}(\mathbf{I}), \text{HOG}_{9\text{x}9}(\mathbf{I}), \tag{4}$$

the decision to use overlapping tiles of 5x5, 7x7, and 9x9 in the FUSION++ method was made to produce a rich and more detailed representation when extracting HOG features. A tile size 5x5 helps capture finer details, and a size 9x9 helps capture broader spatial structures and reduce noise. Lastly, a tile size of 7x7 helps to explore a middle ground between these two extremes.

**Fusion with Attention**: The SW-MHSA blocks help to focus on critical features where manipulations are most likely to happen. These developments in FUSION++ ensure more accurate predictions. In FUSION++, the above features and attention mechanisms are used which can be represented as:

$$\mathbf{F}_{\text{final}} = [\mathbf{F}_{\text{RGB}}, \mathbf{F}_{\text{NP}}, \mathbf{F}_{\text{SRM}}, \mathbf{F}_{\text{BC}}, \mathbf{F}_{\text{HOG}} + A], \tag{5}$$

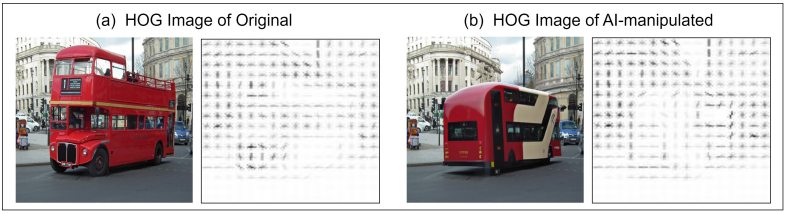


Figure 5: HOG images and their negatives for Original and AI-manipulated Images

where the HOG features and attention map are denoted by $\mathbf{F}_{\text{HOG}}$, and $A$. The output from SW-MHSA blocks is fed to the Cross-Modal Feature Rectification Module (FRM) to refine interactions between features. Then the Feature Fusion Module (FFM) fuses these features into a unified representation.

**Final Detection and Localization**: In this step, the improved features are used for the final detection and localization in FUSION++:

$$S_{\text{final}}, M_{\text{final}} = \text{Detector}(\mathbf{F}_{\text{final}}), \tag{6}$$

where the detection score and manipulation localization map are denoted by $S_{\text{final}}$ and $M_{\text{final}}$.

# 4 Results and Discussions

## 4.1 Experimental Setup

The system used for executing the experiments has a 22-Core GPU Nvidia RTX A4500 and uses the PyTorch framework. In all experimental scenarios, we've used a 70-30% split (8,400 & 3,600 of a total of 12,000) for training and testing from NCL_IMD.v2 dataset that contains original and manipulated images along with its ground truth.

## 4.2 Comparison

In this section we provide a comparison between FUSION++, and several state-of-the-art models [5, 14, 25] on NCL_IMD.v2, AutoSplice, and CocoGlide datasets. the key metrics used for evaluating the detection performance are balanced accuracy (bACC) and area under the curve (AUC). For evaluating the localization performance, we divided the dataset into 4 subsets according to the manipulation percentage in the ground truth (0-25%), (25-50%), (50-75%), and (75-100%). This approach allows us to evaluate the model's performance on different levels of manipulation.

Table 2: Comparison of detection performance using bACC and AUC metrics

| Model | AutoSplice | | CocoGlide | | NCL_IMD.v2 | |
|---|---|---|---|---|---|---|
| | bACC | AUC | bACC | AUC | bACC | AUC |
| MVSS | 0.635 | 0.709 | 0.648 | 0.726 | 0.645 | 0.687 |
| CAT-Net | 0.367 | 0.467 | 0.418 | 0.479 | 0.462 | 0.527 |
| MMFUSION | 0.669 | 0.821 | 0.703 | **0.910** | 0.681 | 0.710 |
| FUSION++ [Our] | **0.680** | **0.849** | **0.752** | 0.885 | **0.937** | **0.953** |

Table 3: Comparison of localization performance using Mean IoU and Pixel-level F1 metrics

| Model | AutoSplice | | CocoGlide | | NCL_IMD.v2 | |
|---|---|---|---|---|---|---|
| | Mean IoU | Pixel-level F1 | Mean IoU | Pixel-level F1 | Mean IoU | Pixel-level F1 |
| **Subset 1 with manipulation percentages from 0 to 25%** | | | | | | |
| MVSS | 0.136 | 0.169 | 0.104 | 0.235 | 0.378 | 0.415 |
| CAT-Net | 0.523 | 0.351 | **0.462** | 0.197 | 0.538 | 0.503 |
| MMFUSION | 0.644 | 0.874 | 0.307 | 0.423 | 0.714 | 0.897 |
| FUSION++ [OUR] | **0.671** | **0.890** | 0.432 | **0.504** | **0.806** | **0.914** |
| **Subset 2 with manipulation percentages from 25 to 50%** | | | | | | |
| MVSS | 0.468 | 0.315 | 0.365 | 0.409 | 0.693 | 0.684 |
| CAT-Net | 0.462 | 0.618 | **0.390** | 0.548 | 0.630 | 0.581 |
| MMFUSION | 0.701 | 0.744 | 0.363 | 0.651 | 0.748 | 0.847 |
| FUSION++ [OUR] | **0.730** | **0.760** | 0.374 | **0.692** | **0.769** | **0.850** |
| **Subset 3 with manipulation percentages from 50 to 75%** | | | | | | |
| MVSS | 0.652 | 0.352 | **0.604** | 0.389 | 0.685 | 0.508 |
| CAT-Net | 0.497 | 0.718 | 0.379 | 0.733 | 0.522 | 0.574 |
| MMFUSION | 0.674 | 0.736 | 0.428 | 0.807 | 0.715 | 0.736 |
| FUSION++ [OUR] | **0.692** | **0.738** | 0.582 | **0.831** | **0.803** | **0.798** |
| **Subset 4 with manipulation percentages from 75 to 100%** | | | | | | |
| MVSS | **0.819** | 0.380 | **0.813** | 0.332 | 0.627 | 0.601 |
| CAT-Net | 0.426 | 0.816 | 0.423 | 0.842 | 0.536 | 0.760 |
| MMFUSION | 0.388 | 0.850 | 0.317 | **0.902** | 0.693 | 0.891 |
| FUSION++ [OUR] | 0.563 | **0.871** | 0.482 | 0.881 | **0.749** | **0.939** |

As depicted in Table 2, for the AutoSplice dataset, FUSION++ produces a bACC of 0.680 and an AUC of 0.849. Whereas MMFUSION produces a bACC of 0.669 and an AUC of 0.821. FUSION++ outperforms MMFUSION by little in both bACC and AUC. For CocoGlide, FUSION++ is better than MMFUSION in bACC. However, AUC is slightly lower for the CocoGlide dataset. The AUC for MFUSION and FUSION++ are 0.910 and 0.885. For NCL_IMD.v2 dataset, FUSION++ outperformed another model by far, which produces a remarkable bACC and AUC0 of 0.937 and 0.953. The overall performance of FUSION++ was consistent with all three datasets for the given performance metrics.

As depicted in Table 3 which contains localization performance (Mean IoU and Pixel-level F1) statistics, FUSION++ outperforms MMFUSION. For the subset (0-25%) manipulation, FUSION++ achieves a better Mean IoU of 0.806 and Pixel-level F1 of 0.914 compared to MMFUSION's 0.714 and 0.897, respectively. For the (25-50%) manipulation subset, FUSION++ performed consistently and produced a better Mean IoU of 0.769 and Pixel-level F1 of 0.850, slightly higher than MMFUSION's 0.748 and 0.847. The trend is consistent throughout with (50-75%) manipulation subset, where FUSION++ performs better than MMFUSION, producing a Mean IoU of 0.803 and Pixel-level F1 of 0.798, compared to MMFUSION's 0.715 and 0.736. For (75-100%) manipulation subset, FUSION++ produces a Mean IoU of 0.749 and an impressive Pixel-level F1 of 0.939, performing again better than MMFUSION, which scores 0.693 and 0.891 in these performance metrics. In Figure 6

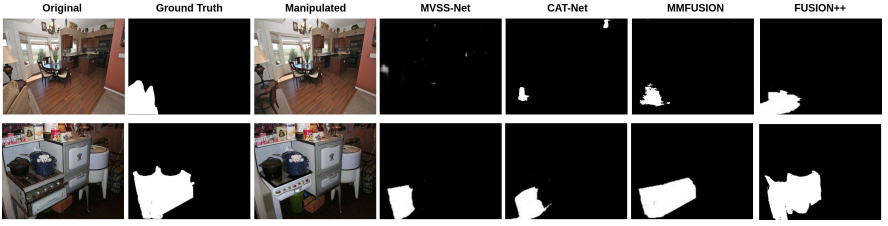


Figure 6: Comparison of qualitative results with current state-of-the-art methods

Table 4: Performance comparison across different configurations using Pixel-level F1 metric

| Configuration | AutoSplice | CocoGlide | NCL_IMD.v2 | AVG | Findings |
|---|---|---|---|---|---|
| MMFUSION | 0.6644 | 0.5636 | 0.8935 | 0.7071 | Reference Point |
| MMFUSION + HOG | 0.6810 | 0.5743 | 0.9052 | 0.7201 | Performance Improved |
| FUSION++ | **0.6817** | **0.6011** | **0.9242** | **0.7356** | Performance Improved |

the qualitative results a clear improvement over state-of-the-art methods. The integration of HOG features and shifted window-based attention mechanism in FUSION++ help to better understand regular and irregular patterns in the images. In all tables, we marked the best results in bold and underlined the second-best results.

### 4.3 Ablation Study

As depicted in Table 4, the addition of HOG features enhances overall feature quality in FUSION++, which is a noticeable improvement. However, FUSION++ further introduction of shifted window-based attention mechanism improves performance. The proof of this addition and updates to MMFUSION are significant to the success of FUSION++.

## 5 Conclusions

This paper introduced two crucial components in detecting image manipulation in the age of generative AI: a) a much-needed AI-manipulated dataset and b) a new detection algorithm. The new dataset was created in an orderly fashion with the use of available generative AI tools and is expected to support the wider research community. The proposed FUSION++ is an advanced model for the detection and localizing of AI-manipulated images. FUSION++ combines the power of HOG features and shifted window-based attention mechanisms. This integration improved performance compared to the state-of-the-art MMFUSION method in terms of detection and localization. The performance of FUSION++ is consistent on multiple datasets and significantly outperforms the state-of-the-art on the new dataset.

# References


[1] Irene Amerini, Lamberto Ballan, Roberto Caldelli, Alberto Del Bimbo, and Giuseppe 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. doi: 10.1109/TIFS.2011.2129512.

[2] Tiziano Bianchi and Alessandro Piva. Image forgery localization via block-grained analysis of jpeg artifacts. *IEEE Transactions on Information Forensics and Security*, 7 (3):1003–1017, 2012. doi: 10.1109/TIFS.2012.2187516.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. ICLR 2019 Conference, 2019.

[4] T. J. De Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. D. R. Rocha. Exposing digital image forgeries by illumination color classification. *IEEE Transactions on Information Forensics and Security*, 8(7):1182–1194, 2013. doi: 10.1109/TIFS.2013.2265677.

[5] Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14185–14193, October 2021.

[6] V. Christlein, C. Riess, J. Jordan, and E. Angelopoulou. An evaluation of popular copy-move forgery detection approaches, 2012.

[7] D. Cozzolino, G. Poggi, and L. Verdoliva. Efficient dense-field copy-move forgery detection. *IEEE Transactions on Information Forensics and Security*, 10(11):2284–2297, November 2015. doi: 10.1109/TIFS.2015.2455334.

[8] G. Fu, Y. Zhang, and Y. Wang. Image copy-move forgery detection based on fused features and density clustering. *Applied Sciences*, 13(13):7528, June 2023. doi: 10.3390/APP13137528.

[9] Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nicholas Dufour, and Luisa 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 (CVPR)*, pages 20606–20615, June 2023.

[10] Yu-Feng Hsu and Shih-Fu 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.

[11] Shan Jia, Mingzhen Huang, Zhou Zhou, Yan Ju, Jialing Cai, and Siwei Lyu. Autosplice: A text-prompt manipulated image dataset for media forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 893–903, June 2023.

[12] K. D. Kadam, S. Ahirrao, and K. Kotecha. Multiple image splicing dataset (misd): A dataset for multiple splicing. *Data (Basel)*, 6(10), October 2021. doi: 10.3390/data6100102.

[13] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[14] Myung-Joon Kwon, In-Jae Yu, Seung-Hun Nam, and Heung-Kyu Lee. Cat-net: Compression artifact tracing network for detection and localization of image splicing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 375–384, 2021.

[15] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence 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.

[17] Xiaohong Liu, Yaojie Liu, Jun Chen, and Xiaoming 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. doi: 10.1109/TCSVT.2022.3189545.

[18] OpenAI. Chatgpt: A large language model. https://chat.openai.com/, 2024. Accessed: 2024-08-12.

[19] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.

[20] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, June 2022. doi: 10.1109/CVPR52688.2022.01042.

[21] V. Schetinger, M. M. Oliveira, R. da Silva, and T. J. Carvalho. Humans are easily fooled by digital images. *Computers and Graphics*, 68:142–151, November 2017. doi: 10.1016/J.CAG.2017.08.010.

[22] Zhihao Sun, Haipeng Fang, Juan Cao, Xinying Zhao, and Danding Wang. Rethinking image editing detection in the era of generative AI revolution. In *ACM Multimedia 2024*, 2024. URL https://openreview.net/forum?id=m83dD4v0SZ.

[23] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022.

[24] Dijana Tralic, Ivan Zupancic, Sonja Grgic, and Mislav Grgic. Comofod — new database for copy-move forgery detection. In *Proceedings ELMAR-2013*, pages 49–54, 2013.

[25] Konstantinos Triaridis and Vasileios Mezaris. Exploring multi-modal fusion for image manipulation detection and localization. In Stevan Rudinac, Alan Hanjalic, Cynthia Liem, Marcel Worring, Björn Þór Jónsson, Bei Liu, and Yoko Yamakata, editors, *MultiMedia Modeling*, pages 198–211, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-53311-2.

[26] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024.

[27] B. Wen, Y. Zhu, R. Subramanian, T. T. Ng, X. Shen, and S. Winkler. Coverage - a novel database for copy-move forgery detection. In *Proceedings - International Conference on Image Processing, ICIP*, pages 161–165. IEEE Computer Society, August 2016. doi: 10.1109/ICIP.2016.7532339.

[28] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18381–18391, 2023.

[29] Markos Zampoglou, Symeon Papadopoulos, and Yiannis Kompatsiaris. Detecting image splicing in the wild (web). In *2015 IEEE international conference on multimedia & expo workshops (ICMEW)*, pages 1–6. IEEE, 2015.

[30] Z. Zhang, Y. Zhou, J. Kang, and Y. Ren. Study of image splicing detection, 2008.

[31] Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. *arXiv preprint arXiv:2312.03594*, 2023.