

MMFusion: Combining Image Forensic Filters for Visual Manipulation Detection and Localization

Kostas Triaridis^{*†}, Konstantinos Tsigos^{*} and Vasileios Mezaris^{*}

^{*}Centre for Research and Technology Hellas (CERTH) / Information Technologies Institute (ITI)

Thermi 57001, Greece, {triaridis, ktsigos, bmezaris}@iti.gr

[†]Department of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA

ktriaridis@cs.stonybrook.edu

Abstract—Recent image manipulation localization and detection techniques typically leverage forensic artifacts and traces that are produced by a noise-sensitive filter, such as SRM or Bayar convolution. In this paper, we showcase that different filters commonly used in such approaches excel at unveiling different types of manipulations and provide complementary forensic traces. Thus, we explore ways of combining the outputs of such filters to leverage the complementary nature of the produced artifacts for performing image manipulation localization and detection (IMLD). We assess two distinct combination methods: one that produces independent features from each forensic filter and then fuses them (this is referred to as late fusion) and one that performs early mixing of different modal outputs and produces combined features (this is referred to as early fusion). We use the latter as a feature encoding mechanism, accompanied by a new decoding mechanism that encompasses feature re-weighting, for formulating the proposed MMFusion architecture. We demonstrate that MMFusion achieves competitive performance for both image manipulation localization and detection, outperforming state-of-the-art models across several image and video datasets. We also investigate further the contribution of each forensic filter within MMFusion for addressing different types of manipulations, building on recent AI explainability measures.

Index Terms—Image forensics, Image manipulation localization, Image manipulation detection, Video manipulation detection, Noise-sensitive filters, Multi-modal fusion

I. INTRODUCTION

Editing and manipulating digital media has gotten increasingly easier and more accessible in recent years. Recent advances in image editing software, as well as deep generative models such as Generative Adversarial Networks (GANs) [1], [2] and diffusion models [3], [4], facilitate producing manipulations that are often imperceptible to the human eye and are widely available, even to potentially malicious users. The widespread use of smartphones and social networks also enables the spread of such manipulated media at a rapid pace. As a result, such edited images can cause social problems when used as evidence to support disinformation campaigns and stories or mislead the public by obfuscating important content from news, resulting in diminished trust. Therefore, techniques for image manipulation detection and localization, as part of complete toolboxes for media verification such as [5], are now needed more than ever.

Image forgery localization and detection are tasks the media forensics field has been working on for many years. Early

works typically focused on a specific type of manipulation such as splicing [6], copy-move [7] or removal/inpainting [8]. More recently, deep-learning-based solutions of increasing robustness are proposed that are able to recognize multiple different types of manipulations [9]–[16]. In order to be able to perform manipulation localization in a semantic-agnostic manner, these models need to suppress image content to reveal forensic artifacts. Most approaches achieve this by applying a high-pass filter to extract noise maps [9], [12], [14]–[16]. The most popular high-pass filters used are the ones proposed in the Steganalysis Rich Model (SRM) [17], utilized in a wide variety of works [12], [15], [18]–[20], while the Bayar convolution [21] is also used in a multitude of approaches [9], [12], [14], [15] and NoisePrint is used in a more recent model [11].

We hypothesize that those different forensic filters actually produce artifacts of complementary forensic capabilities. NoisePrint [22] and its successor NoisePrint++ [11] produce artifacts that relate to camera model and editing history, thus displaying limited performance for copy-move manipulations (see Sec. IV-C for results supporting this statement). On the other hand, SRM [17] filters can identify edges and boundaries without relying on camera or compression/editing artifacts. These filters are, however, fixed by design (not trainable or changeable), a property that makes them vulnerable to adversarial attacks; whereas the Bayar convolution [21], on the other hand, learns the manipulation traces directly from data, proving more robust against malicious attacks. In this work we explore ways to expand existing state-of-the-art Image Manipulation Localization and Detection (IMLD) approaches to support multiple forensic filters as inputs. We start with TruFor [11] as our baseline and propose utilizing NoisePrint++, SRM, and Bayar convolution as inputs auxiliary to the RGB image. We initially assess two different approaches to feature encoding: a late-fusion paradigm that extracts and encodes features from each modality (filter) separately, and an early-fusion paradigm that mixes the multi-modal features by early convolutional blocks. We also improve the decoder architecture of [11] by introducing feature re-weighting (in both the Anomaly and Confidence decoders), improving the model’s capability for recognizing and localizing anomalies. We then propose the MMFusion architecture (Fig. 1), which employs early fusion and the aforementioned decoder architecture, and we also explore ways of explaining the predictions and understanding the unique capabilities of the different forensic

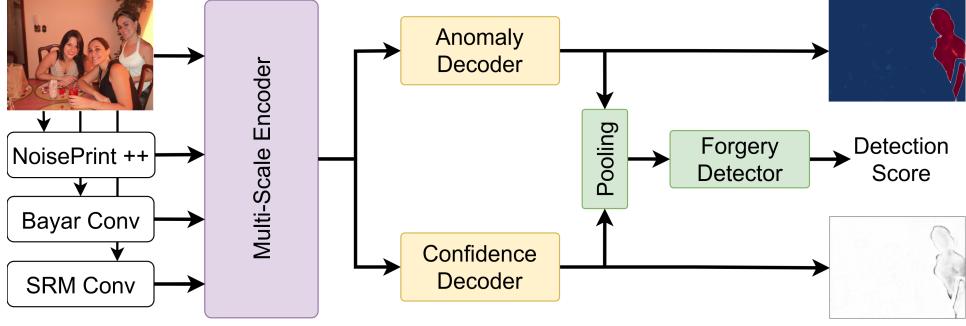


Fig. 1: Overview of the MMFusion Encoder-Decoder architecture for image localization and detection with multiple forensic filters. The RGB image and the output of each filter are fed into a Multi-Scale encoder, whose output is passed onto both the anomaly decoder, which produces a localization map, and the confidence decoder, which produces a confidence map. The two maps are then combined through a pooling module and passed into the forgery detector to produce the manipulation detection score.

filters of MMFusion. We find that the different filters excel at recognizing different kinds of image manipulations, validating our original hypothesis regarding their complementarity.

Furthermore, we investigate the capabilities of IMLD models without a special temporal-aware architecture for detecting and localizing manipulations in videos. We demonstrate that by using frame-level predictions with our MMFusion IMLD model we can reach state-of-the-art performance in Video Manipulation Localization and Detection (VMLD) tasks, documenting the merits of MMFusion while also highlighting the possibly limited complexity of the existing VMLD benchmarks.

A preliminary version of this work, with a simpler decoder architecture (without the proposed Feature Re-weighting Decoder) and without the studies on the explainability of multimodal IMLD models and the applicability of IMLD models to VMLD tasks, was presented in [23]. Our main contributions in this paper are summarized as follows:

- We compare the efficacy of different forensic filters, namely SRM, Bayar convolution and NoisePrint++, as inputs for deep networks performing forgery localization.
- We assess two distinct approaches for combining the outputs of different forensic filters for the purpose of image manipulation localization and detection, and we propose the MMFusion architecture.
- We propose, as part of the MMFusion architecture, a new Feature Re-weighing Decoder (FRD) that significantly increases localization performance.
- We also investigate the applicability of IMLD models, and our model specifically, on video datasets and compare it to state-of-the-art Video Manipulation Localization and Detection models. We compare our performance both with models specifically designed to tackle temporal inconsistencies for VMLD and with simple IMLD models, and we provide a new baseline for the application of IMLD models on video without architectural changes.
- We propose a method for explaining the predictions of our multi-modal IMLD model by quantifying the contribution of each forensic filter for a specific image. This enables us to investigate the efficacy of each filter

for different manipulation types and thus provide a deeper understanding of their predictive capabilities.

II. RELATED WORK

A. Image Manipulation Localization and Detection

Traditional image forensics methods, e.g. [24]–[26], have largely focused on detecting inconsistencies in low-level semantic-agnostic compression and internal camera artifacts. These artifacts can usually be revealed through high-pass filtering techniques that produce a noise-sensitive visualization of the image.

In recent times, various filters for noise extraction have been integrated into deep learning models to address the challenge of image manipulation localization and detection. Zhou et al. proposed RGB-N [18], a two-stream Faster R-CNN network [27] that utilizes the RGB channel features as well as the extracted SRM-based noise features of the input image to detect visual inconsistencies and identify mismatches between authentic and tampered regions, originating from image splicing, to perform forgery detection with bounding boxes. In a similar fashion, Yang et al. showcased Constrained R-CNN (CR-CNN) [14], a coarse-to-fine end-to-end architecture, which uses a learnable manipulation feature extractor (LMFE) based on a Bayar convolution, to create a unified feature representation for various manipulation types directly from the data. CR-CNN then follows two distinct stages: stage 1 performs manipulation technique classification and coarse manipulated region localization using the attention regional proposal network (RPN-A), while stage 2 fuses low- and high-level information to refine the global manipulation features. The model finally combines these refined features with the coarse localization information to further learn the finer local features and perform tampered region segmentation. Wu et al. [12] conducted experiments with various backbone network architectures and feature choices for proposing ManTraNet, a VGG-based manipulation localization and detection model, that integrates both SRM filters and Bayar convolution and is composed of three stages: adaptation, which adapts the manipulation trace feature for the anomaly detection task; anomalous feature extraction, which is inspired by human

thinking and extracts anomalous features; and decision, which holistically considers anomalous features and classifies each pixel as either forged or not. Hu et al. presented SPAN (Spatial Pyramid Attention Network) [15], a framework for detecting and localizing various image manipulations by utilizing a hierarchical pyramid structure that models and encodes the relationships and the spatial positions of image patches at multiple scales using local self-attention blocks and position projection. It includes three blocks: a feature extractor, a spatial pyramid attention block, and a decision module applied on top of the output from the spatial pyramid attention propagation module to predict the localization mask. Moreover, Chen et al. introduced MVSSNet [9], a network that fuses the features from a ResNet-based edge-supervised branch (with a Sobel layer for edge enhancement) and a noise-sensitive branch using Bayar convolution via a trainable Dual Attention (DA) module in a late fusion paradigm, trained through multi-scale supervision. To combat the limited generalizability of MantraNet and SPAN, due to them being unable to fully take advantage of the spatial correlation, Liu et al. [10] proposed PSCC-Net, which employs a Spatio-Channel Correlation Module (SCCM) that leverages the Gaussian function, to capture spatial and channel-wise correlations. PSCC-Net is a two-path structure architecture that follows a top-down path for extracting local and global features and a bottom-up path for detecting manipulations and estimating manipulation masks at multiple scales. It then uses dense cross-connections to fuse features across scales in a coarse-to-fine manner. Extending on their previous work that only targeted splicing forgery, Kwon et al. [13] expanded their research to additionally deal with copy-move forgery and introduced CatNetv2, which captures image acquisition camera-specific artifacts in the RGB domain and compression artifacts in the DCT domain and localizes the manipulated regions by considering the domains jointly. Similarly, Wang et al. defined ObjectFormer [16], an end-to-end multi-modal framework that combines RGB features and frequency features and consists of a High-frequency Feature Extraction Module, an object encoder that uses learnable object queries to learn whether mid-level representations in images are coherent, and a patch decoder that produces refined global representations for manipulation detection and localization. Shi et al. [28] also proposed a dual domain-based CNN architecture with a spatial-domain CNN model (Sub-SCNN), that utilizes SRM filters and performs hierarchical feature extraction, and a frequency domain-based CNN model (Sub-FCNN) that extracts statistical features using the 3-level Daubechies-based Discrete Wavelet Transformation (DWT). Song et al. [29] created the Tri-Path Backbone Architecture (TPB-Net) that consists of three DenseNet169 networks in a feature pyramid structure, to integrate features from different levels. They introduce a Dual-path Compressed Sensing Attention (DCSA) module to facilitate feature fusion, with the reasoning that high-level feature maps generally contain richer semantic information. Focusing specifically on inpainting region localization, Daryani et al. [30] defined a CNN-based deep learning model called IRL-Net, which includes three main modules: the enhancement module that tries to enhance inpainting traces with a Bayar layer, the encoder which

includes four residual units to avoid vanishing/exploding gradients, and a Decoder that uses attention to map the learned high-level features extracted by the encoder. Finally, Guillaro et al. leveraged NoisePrint [22], a noise extractor proposed by Cozzolino et al. that is trained in a self-supervised manner to extract camera-specific artifacts and expanded its use in TruFor [11], where it is used jointly with RGB images in a dual-branch CMX [31] architecture.

Contrary to most of the above works, that rely on a single forensic filter, our approach innovatively explores strategies for combining the outputs of three diverse noise extractors, leveraging their complementary capabilities to develop a robust end-to-end image forgery detection and localization model.

B. Video Manipulation Localization and Detection

Early works on video forensics, much like image forensics, relied heavily on non-learning based signal processing techniques to extract forensic artifacts. These methods were generally restricted in the types of forgeries they could detect and in the accuracy of said detection [32].

In recent years, the trend has shifted toward deep learning-based approaches, enabling the development of networks capable of recognizing a broad range of forgeries. However, most methods have focused on the detection of a narrow set of very specific forgeries [33], such as frame insertion and deletion [34]–[36], or most commonly face deepfakes [20], [37]–[39]. Very few approaches have tackled the general problem of video forgery detection directly. The advent of models like VideoFACT [32], [40] illustrates the shift toward deep learning-based frameworks. VideoFACT [40] incorporates forensic and contextual embeddings to capture traces left by manipulation and check for variations in forensic traces introduced by video coding. Subsequently, to estimate the quality and the relative importance of these local embeddings, it employs a deep self-attention mechanism.

Similarly to most works on IMLD, the few existing VMLD methods do not leverage multiple forensic filters and their potential complementarity.

III. METHODOLOGY

A. Encoder-Decoder Architecture

Our goal is to extend an existing encoder-decoder-based architecture to be able to use multiple forensic filters (SRM [17], Bayar convolution [21], NoisePrint++ [11]) in tandem, so as to produce more robust representations for the IMLD task. To this end we adopt the general architecture of TruFor [11], i.e., as illustrated in Fig. 1 we use an encoder, an anomaly decoder, a confidence decoder, and a forgery detector; and we follow TruFor’s two-phase training regime for anomaly localization and detection, respectively. The encoder follows the popular dual-branch architecture proposed in [31] and illustrated in Fig. 2 for a single forensic filter, comprising of 4 stages of Multi-Head Self Attention (MHSA) blocks [41] that produce feature maps f_{mod}^i of different scales: $\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times C_i$, where $i \in \{1, 2, 3, 4\}$, $mod \in \{\text{image}, \text{filter}\}$, H and W are the spatial dimensions of the input image and C_i is the channel dimension of the output at stage (and scale) i . The two MHSA

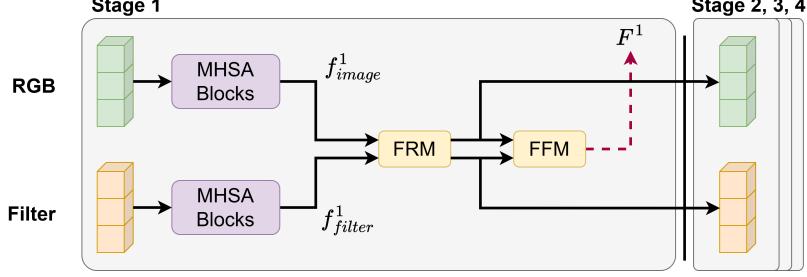


Fig. 2: Architecture of the dual-branch encoder of [31]. The encoder is made of 4 stages of Multi-Head Self Attention (MHSA) blocks to produce feature maps f_{mod}^i for modality $mod \in \{image, filter\}$ and stage $i \in \{1, 2, 3, 4\}$. These are then fused and rectified by the FRM and FFM modules to produce the outputs F^i at each scale i . The feature map set $F = \{F^i, i = 1, \dots, 4\}$ is the final output returned by the encoder.

TABLE I: Main symbols used in Sec. III for denoting feature maps.

Symbol	Description
f_{mod}^i	Feature maps returned by the MHSA block of the encoder for modality $mod \in \{image, filter\}$ at stage i (Fig. 2)
F^i	Fused feature maps returned by the FFM module of the encoder at stage i (Fig. 2, 4)
f_{filter}^i	Feature maps returned by the MHSA block of the late fusion encoder for filter $filter \in \{\text{noiseprint}, \text{srn}, \text{bayar}\}$ at stage i (Fig. 3)
F_{filter}^i	Feature maps returned by the FFM module of the late fusion encoder for filter $filter \in \{\text{noiseprint}, \text{srn}, \text{bayar}\}$ at stage i (Fig. 3)
f_a	Mixed features produced by the early fusion module (Fig. 4)
$F = \{F^i\}$	Output feature maps of the encoder (Fig. 5)
$D = \{D^i\}$	Re-weighted feature maps fed to the MLP-based decoder (Fig. 5)

blocks' outputs (for the RGB image and the filter) in each stage are rectified through a Cross-Modal Feature Rectification Module (FRM) [31] that exploits the interactions between the two input modalities (RGB and NoisePrint++ in the case of TruFor). The FRM uses features from both modalities to produce weighted channel- and spatial-wise feature maps that are residually added for both modalities to perform channel- and spatial-wise rectification. The two sets of feature maps are then combined using a Feature Fusion Module (FFM) [31], whose outputs F^i (having the same dimensions as the outputs of the MHSA blocks at each scale i) for $i \in \{1, 2, 3, 4\}$ collectively constitute the encoder output $F = \{F^i, i = 1, \dots, 4\}$ (see Table I for notation summary, and references to the relevant architecture diagrams). The FFM consists of an information exchange stage, where a cross-attention mechanism exchanges information between modalities and produces two sets of mixed feature maps, and a fusion stage where the feature maps are merged into a single output through a residual Multilayer Perceptron (MLP) module that uses 1×1 convolutions. The Decoders illustrated in Fig. 1 are, in the case of TruFor [11], MLP-based decoders proposed in [41].

Utilizing this architecture one can combine RGB images with an auxiliary forensic modality to perform Image Manipulation Localization. In [11] Guillaro et al. use their own feature extractor NoisePrint++, however a multitude of other forensic filters' outputs, such as Bayar convolution [21] or SRM [17],

can be utilized. These filters are analyzed in Sec. III-B. We assess two different ways of extending the encoder architecture to multiple auxiliary modal inputs: a late fusion paradigm, where each auxiliary modality is combined with RGB inputs separately using a dual-branch architecture [31] (Sec. III-C), and an early fusion paradigm where auxiliary modalities are combined early before being utilized as input to the dual-branch encoder together with the RGB inputs (Sec. III-D). We then propose the MMFusion architecture, that extends the encoder architecture to multiple auxiliary modal inputs using early fusion and introduces a feature re-weighting step in the decoders.

B. Auxiliary (a.k.a. filter) modalities

For both early- and late-fusion approaches, we use the outputs of three forensic filters, namely NoisePrint++, SRM and Bayar convolution, as inputs that are auxiliary to the RGB image. We choose these filters as they are widely used in the relevant literature (Sec. II-A), they showcase good performance and also complement each other well: SRM is a static feature extractor that mostly extracts edge features, while NoisePrint++ and Bayar convolution are trainable modules that are, however, trained with different objectives. NoisePrint++ is trained in a self-supervised contrastive manner as a camera "fingerprint" extractor, while Bayar convolution is directly trained for IMLD in a supervised setting, as explained below.

1) *NoisePrint++*: In [22] Cozzolino et al. proposed NoisePrint, a CNN-based model designed to extract camera-model-based artifacts from RGB images while suppressing image content. In [11] they expanded their approach, namely NoisePrint++, to be able to recognize and extract artifacts related to the editing history of an image (e.g. compression, resizing, gamma correction). NoisePrint++ is trained in a supervised contrastive manner [42]: a batch of images is provided, from which patches are extracted from different locations. Then the patches go through different editing pipelines. Patches extracted from the same source image, the same location, and with the same editing history are considered positive samples, while others are considered negative. In our work we use NoisePrint++ as a pretrained feature extractor.

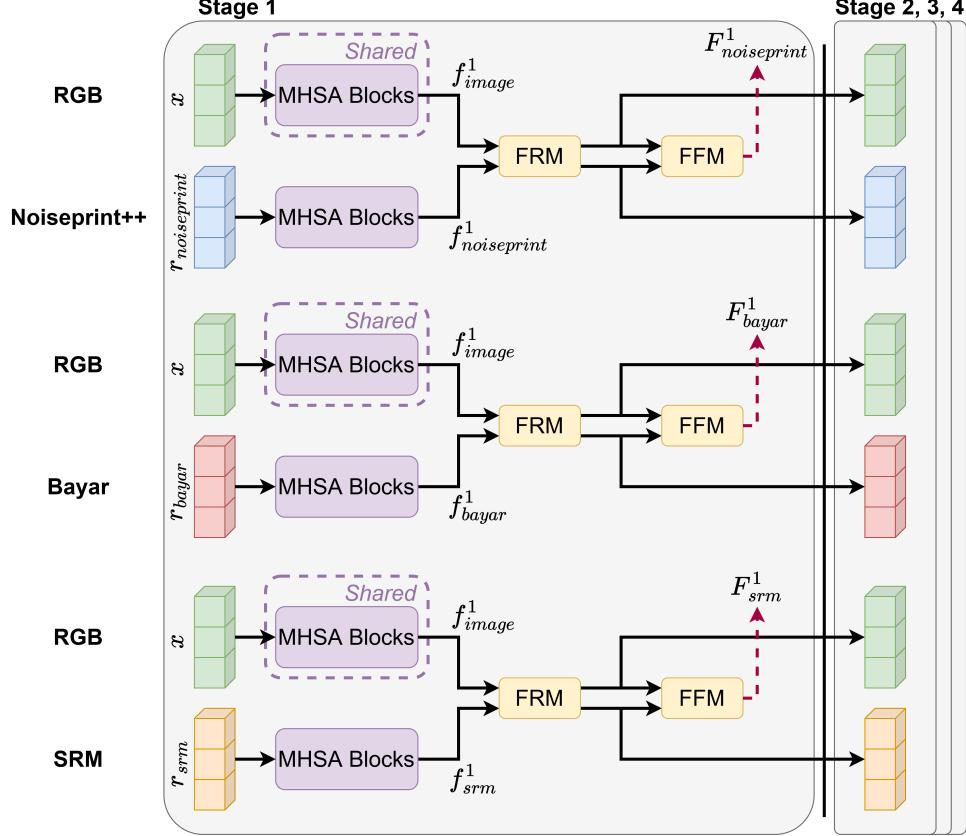


Fig. 3: Proposed architecture of the encoder for fusion of multiple forensic filters by late fusion with weight sharing. The filters’ outputs and the RGB image are fed into separate MultiHead Self-Attention (MHSA) blocks of the dual-branch CMX encoder, with the outputs rectified and combined by the FRM and FFM modules to produce the feature maps. These are propagated through different stages to create feature maps of varying scales. The weights of the MHSA blocks of all RGB branches are shared to increase regularization.

2) *SRM*: Another way to suppress the image content and highlight forensic traces and noise is through static high-pass filters, the most common of which are the ones proposed for producing residual maps for the Steganalysis Rich Model (SRM) [17]. Out of the 30 high-pass filters proposed in [17], we use the 3 most commonly used in the literature, e.g. as in [12], [15], [18], which are displayed in Fig. 4 of [18]. Following [17] and [18], the outputs of these three filters are truncated and combined to form the final noise descriptor, referred in the sequel as SRM filter.

3) *Bayar Convolution*: In contrast to using static high-pass filters for noise extraction, Bayar et al. [21] proposed the constrained convolutional layer as a noise extractor that adaptively learns manipulation traces from data. We use the constrained convolutional layer as an extra noise feature extractor and refer to it as Bayar convolution. For both our multi-modal fusion approaches the Bayar convolutional layer is pretrained alone in a dual branch CMX encoder [31] (as also done in our ablation study for examining the effect of using Bayar as the sole auxiliary input alongside the RGB image; see Sec. IV-C and the results for ‘‘CMX (Bayar)’’ in Table VI) and then used with its weights frozen.

C. Late Fusion

For the late fusion approach we extract the auxiliary representations $r_{noiseprint}$, r_{srmb} , r_{bayar} of the RGB image x from the NoisePrint++, SRM and Bayar filters respectively. Then the output of each filter is fed together with the original RGB input into a dual-branch CMX encoder \mathcal{E} , made of 4 Multi-Head Self Attention (MHSA) blocks [41] that produce 4-scale feature maps f_{mod}^i , where $mod \in \{\text{image}, \text{filter}\}$, $\text{filter} \in \{\text{noiseprint}, \text{srmb}, \text{bayar}\}$ and $i \in \{1, 2, 3, 4\}$. These feature maps are passed to and rectified through the FRM and FFM modules, to produce the final features of the encoder $F_{\text{filter}}^i = \mathcal{E}_{\text{filter}}(x, r_{\text{filter}})$ as shown in Fig. 3.

The outputs F_{filter}^i of the three encoders for a given i are concatenated, and the resulting set of feature maps for $i \in \{1, 2, 3, 4\}$ constitutes the final output F of the encoder, which is then passed to the decoders (as illustrated in Fig. 1). In this late fusion approach we use the same decoder architecture as in TruFor for both the anomaly and confidence decoders. Like other multi-modal approaches, this approach is prone to overfitting and the ‘‘modality imbalance’’ problem [43], [44], where different modalities converge and overfit at different rates, thus hindering joint optimization. To tackle this we make the weights of the modules along the RGB branch

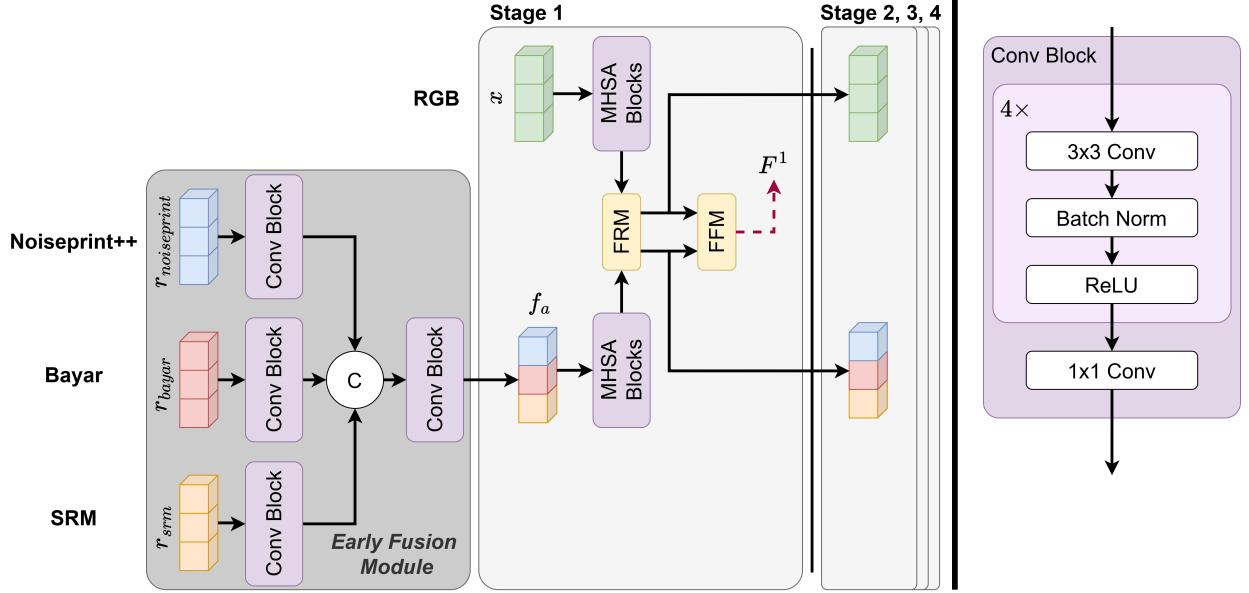


Fig. 4: Proposed architecture of the encoder for fusion of multiple forensic filters by early convolutions. On the left, we illustrate the structure of the encoder. More specifically, the filters' outputs are initially fused by early convolutional blocks in the Early Fusion Module, to produce the mixed features f_a . These features and the RGB image are then fed into separate MultiHead Self-Attention (MHSA) blocks of a dual-branch CMX encoder, with the outputs rectified and combined by the FRM and FFM modules to produce the feature maps. These are propagated through different stages to create feature maps of varying scales. The structure of the convolutional block is presented on the right side.

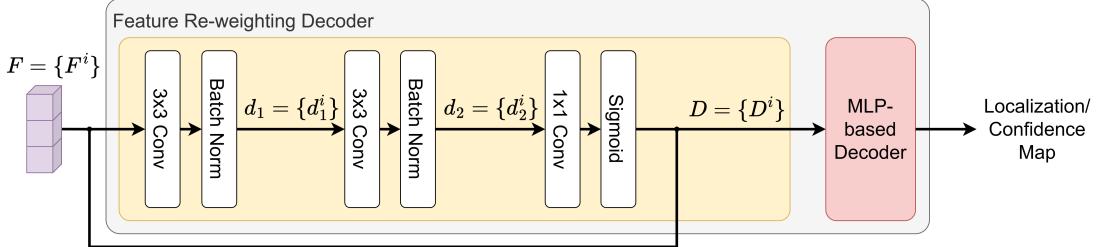


Fig. 5: Proposed architecture of the Feature Re-weighting Decoder (FRD). The feature maps F returned from the encoder are processed through convolutional layers, batch normalization and activation functions, and weighted channel- and spatial-wise feature maps that enhance subtle variations in the input maps are produced. These are then passed to the MLP-based decoder to generate the localization/confidence map.

shared across all 3 encoders to increase regularization. We also employ Dropout before the anomaly decoder as the complete encoder is rather large and the simple MLP-based decoder is prone to overfitting.

D. Early Fusion

For the early fusion approach we extract again the same auxiliary representations $r_{noiseprint}, r_{srm}, r_{bayar}$ of the RGB image x . The inputs are then passed through our novel Early Fusion Module \mathcal{EFM} to produce the mixed features $f_a = \mathcal{EFM}(r_{noiseprint}, r_{srm}, r_{bayar})$ as shown in Fig. 4. The \mathcal{EFM} consists of 3 independent convolutional blocks, one for each auxiliary modality, and one final convolutional block that performs feature mixing. The convolutional blocks are good at early visual processing, resulting in a more stable optimization [45], thus aiding in mixing the features from different modalities smoothly. The mixed features f_a and

RGB image x are used as input for a dual-branch CMX encoder [31], in the same manner as in TruFor. This is a particularly lightweight approach to expanding the TruFor architecture to handle multiple auxiliary modalities, as it does not significantly increase the number of parameters of the network (68.9M params compared to TruFor's 68.7M, as reported in [11]).

Each of the convolutional blocks mentioned above consists of four 3×3 convolutions followed by a 1×1 convolutional layer to resize the output to 3 channels (Fig. 4). There is a batch normalization (BN) and a ReLU layer after each 3×3 convolutional layer. The output channels for the 3×3 convolutional layers are [24, 48, 96, 192].

E. Feature Re-weighting Decoder (FRD)

For the decoder part of our network we enhance the MLP-based decoder used in CMX [31] to implement both the

anomaly and the confidence decoder (see Fig. 1), with feature re-weighting. The motivation behind this is that, for harder-to-recognize manipulations, the encoder may produce feature maps that are not dissimilar enough between the image’s original and manipulated regions. Thus we employ feature re-weighting to enhance the differences between these parts. The process works by taking the feature maps F^i of the feature map set F produced by the encoder and re-weighting them (resulting in re-weighted feature maps D^i), before feeding the multi-layer feature map set $D = \{D^i, i = 1, \dots, 4\}$ into an MLP-based decoder as shown in Fig. 5. The latter produces the final decoder output. This re-weighting ensures that the subtle variations become more pronounced, making it easier for the subsequent MLP-based decoder to differentiate and accurately process the manipulated content. This significantly enhances the network’s ability to detect difficult-to-recognize manipulations. The re-weighted feature maps $D = \{D^i, i = 1, \dots, 4\}$ are calculated (Fig. 5) as follows:

$$\begin{aligned} d_1^i &= BN(Conv_{3 \times 3}(F^i)) \\ d_2^i &= BN(Conv_{3 \times 3}(d_1^i)) \\ D^i &= \text{sigmoid}(Conv_{1 \times 1}(d_2^i)) * F^i \end{aligned}$$

where BN denotes Batch Normalization and $Conv_{3 \times 3}$ and $Conv_{1 \times 1}$ are convolutions with 3×3 and 1×1 kernels, respectively.

IV. EXPERIMENTS

A. Experimental Setup

1) *Training*: We follow the training procedure proposed by Guillaro et al. [11]: first, we jointly train the encoder and anomaly decoder; after that, we train the confidence decoder and the forgery detector, while the encoder and anomaly decoder are kept frozen. For both training phases in IMLD we use the datasets used by Kwon et al. [13], and sample an equal number of images from each one for every epoch. For the VLMD task, we use the datasets proposed in [40]. The employed training datasets are summarized in Table II.

2) *Testing*: For testing, we evaluate our model on five IMLD datasets: Coverage [46], Columbia [47], Casiav1+¹ [49] and DSO-1 [50], which are widely used in the relevant literature, and CocoGlide [11]. The latter is a diffusion-based manipulation dataset proposed recently by Guillaro et al [11], that uses the COCO validation dataset, [51] an object mask with its corresponding label as the forgery region and the text prompt, and feeds them to GLIDE [3] to generate new synthetic objects. As for the other aforementioned datasets, Coverage accommodates copy-move forgery manipulations, while Columbia and DSO-1 focus on splicing manipulations. Finally, Casiav1+ includes a variety of image manipulations such as splicing, copy-move, and removal.

We also evaluate our models on VCMS, VPVM and VPIM [40], which are datasets for Video Manipulation Localization and Detection. VCMS contains splicing manipulations, while

¹Casiav1+ is a modification of the Casiav1 dataset proposed by Chen et al. [9] that replaces authentic images that also exist in Casiav2 with images from the COREL [48] dataset to avoid data contamination.

Dataset	Number of Images	
	Real	Manipulated
Casiav2 [49]	7.491	5.105
IMD2020 [52]	414	2.010
FantasticReality [53]	16.592	19.423
cm_coco [13]	-	200.000
bcm_coco [13]	-	200.000
bcmc_coco [13]	-	200.000
sp_coco [13]	-	200.000
VCMS [40]	48.000	48.000
VPVM [40]	48.000	48.000
VPIM [40]	48.000	48.000

TABLE II: Number of real and manipulated images (or video frames) in each training dataset. (Image datasets above line, video datasets below line)

Dataset	Number of Images	
	Real	Manipulated
Coverage [46]	100	100
Columbia [47]	183	180
Casiav1+ [49]	800	921
DSO-1 [50]	100	100
CocoGlide [11]	512	512
VCMS [40]	300	300
VPVM [40]	300	300
VPIM [40]	300	300

TABLE III: Number of real and manipulated images (or video frames) in each test dataset. (Image datasets above line, video datasets below line)

VPVM and VPIM contain manipulations made with standard video editing operations (blurring, gamma adjustment etc), utilizing different strengths that make them visually perceptible (in VPVM) or imperceptible (in VPIM), respectively. For all video datasets, we sample just the first frame of each video to evaluate our method (hence the number of frames in Table III is 300 for each of the manipulated / non-manipulated classes, equal to the number of test videos for each class that we retrieved from ²). We did such a radical frame sampling because in early experiments we observed that using just one frame per video yielded the same results as averaging across all video’s frames (± 0.01), while naturally requiring significantly less runtime.

The employed testing datasets are summarized in Table III.

3) *Evaluation Measures*: For localization performance we follow most previous works, e.g. [9]–[16], [18], [28]–[30], and report average pixel-level performance using the F1 measure, which uses the ground truth and the prediction mask to determine the True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). The F1 and the inverse F1 scores are then computed and the maximum value between the two is returned. TP are the pixels where the ground truth and the prediction mask overlap to correctly identify manipulated regions, TN are the pixels where the ground truth and the prediction mask overlap to correctly identify non-manipulated regions, FP are the pixels where the prediction incorrectly identifies manipulated regions not present in the ground truth and FN are the pixels where the prediction mask fails to identify manipulated regions present in the ground truth. We use a fixed threshold of 0.5, where pixels

²<https://huggingface.co/datasets/ductai199x/video-std-manip>

TABLE IV: Average pixel-level F1 scores for the localization task on the considered models and image datasets. Best (higher) scores in bold and second best scores underlined. Results for all models except for the proposed ones are taken from [11].

Model	Coverage	Columbia	Casiav1+	CocoGlide	DSO-1	AVG
TruFor [11]	0.600	0.859	0.737	0.523	0.930	0.729
CAT-Netv2 [13]	0.381	0.859	0.752	0.434	0.584	0.602
ManTraNet [12]	0.317	0.508	0.180	0.516	0.412	0.387
PSCC-Net [10]	0.473	0.604	0.520	0.515	0.458	0.514
SPAN [15]	0.235	0.759	0.112	0.298	0.233	0.327
CR-CNN [14]	0.391	0.631	0.481	0.447	0.289	0.448
MVSS-Net [9]	0.514	0.729	0.528	0.486	0.358	0.523
Late Fusion (Sec. III-C)	0.641	0.864	0.775	<u>0.574</u>	<u>0.899</u>	0.751
Early Fusion (Sec. III-D)	<u>0.663</u>	0.888	<u>0.784</u>	0.553	0.863	0.750
MMFusion (Early Fusion with FRD)	0.700	<u>0.876</u>	0.794	0.591	0.866	0.765

with higher value indicate the predicted manipulated regions, as setting a best threshold per test dataset [13] or even per image [11], like some other previous works have done, is not realistic in practical scenarios where the ground truth is not available, thus leading in exaggerated performance estimates. For detection, similarly to e.g. [9]–[12], [14]–[16], [18], [29], we calculate the image-level Area Under Curve (AUC), which is a measure that evaluates the model’s ability to separate the two classes (manipulated and or not) over various thresholds. AUC takes values in the range [0.5,1.0], where a perfect model would score 1.0, while a randomly guessing one would score 0.5. We also employ balanced accuracy (bAcc) as in [11], which is the arithmetic mean of sensitivity and specificity, with the threshold set once again to 0.5.

4) *Implementation:* All models are implemented in PyTorch and trained on a single consumer-grade NVIDIA GPU (either an RTX 4090 or an RTX 3090), using an effective batch size of 24 for 100 epochs. Physical batch size ranged from 4 to 8 depending on the model and an effective batch size of 24 was reached by utilizing gradient accumulation. We use a Dropout rate of 0.3 for the methods proposed in this paper. The MHSA modules were initialized with ImageNet-pretrained weights as proposed in [31], [54]. We utilized an SGD optimizer with an initial learning rate of 0.005, momentum of 0.9, weight decay of 0.0005 and a polynomial learning rate schedule. For training augmentations we followed the protocol of Guillaro et al. [11]: resized the images in the [0.5–1.5] range, performed random cropping of size 512×512 and JPEG compression with a random Quality Factor $QF \in [30,100]$.

B. Evaluation and Comparisons on Image Manipulation Datasets

We compare our methods with recent state-of-the-art approaches for IMLD. Following Guillaro et al. we consider methods with open source models provided and we exclude models that use part of our testing datasets for training to avoid bias. Overall, we compare with TruFor [11], CAT-Netv2 [13], ManTraNet [12], PSCC-Net [10], SPAN [15], Constrained R-CNN [14], MVSS-Net [9]. Results are presented in Table IV.

Both of our multi-modal fusion approaches, as well as the proposed MMFusion architecture, showcase state-of-the-art performance, being either the best or second-best model for every dataset. We also observe that the proposed decoder employed in MMFusion further increases the average F1 by 1.5%, reaching a value of 76.5%. Especially for the Coverage

dataset that contains only copy-move forgeries, MMFusion surpasses the previous best, TruFor, by 10%. The only dataset where we do not achieve top performance is DSO-1, where our best approach (Late Fusion) is 3% behind TruFor.

We also compare across models in terms of detection performance and present the results in Table V. Notably, our early fusion variant and the MMFusion architecture demonstrate exceptional performance, surpassing the state-of-the-art on average. Particularly noteworthy is the outstanding performance on the Coverage dataset, where they achieve a remarkable improvement of nearly 7% in terms of the Area Under the Curve (AUC) and 9% in terms of balanced accuracy (bAcc) compared to the prior leading method. Our late fusion approach also exhibits competitive AUC performance, but falls slightly behind the TruFor model in terms of bAcc. This disparity in bAcc performance could potentially be attributed to the size of our late fusion model, which makes it susceptible to overfitting.

Finally, the effectiveness of our proposed approaches is illustrated with qualitative results in Fig. 6, where we see that all our fusion approaches can approximately localize the existing manipulation(s), with MMFusion achieving more accurate and complete localization. More specifically, in the first image (coming from the Coverage dataset), MMFusion most closely matches the Ground Truth, while the Early Fusion model without FRD over-predicts, highlighting the top of the original glass, and Late Fusion misses key areas like the ribbon. In the second image (from the CocoGlide dataset), MMFusion again performs the best, accurately capturing most of the manipulated region, while the other two fusion approaches leave gaps in the shape of the bus or even fail to detect it all together. For the third image (from the DSO-1 dataset), MMFusion provides the most accurate prediction of the three, as it correctly detects the lower right portion of the shirt as manipulated.

C. Ablation Study

In this section, for the purpose of contrasting the employed forensic filters (SRM, Bayar conv, NoisePrint++), we train a dual-branch CMX architecture where each filter’s output serves as the single auxiliary input alongside the RGB image. The outcomes are presented in Table VI, along with the number of parameters (in millions) and runtime (for a single image on an RTX 3090 GPU) for all methods. During this training the Bayar convolutional layer is trainable, while SRM and

TABLE V: Results of the Area Under Curve (AUC) and balanced accuracy (bAcc) measures for the detection task on the considered models and image datasets. Best (higher) scores in bold and second best scores underlined.

Model	Coverage		Columbia		Casiav1+		CocoGlide		DSO-1		AVG	
	AUC	bAcc										
TruFor [11]	0.770	0.680	0.996	0.984	0.916	0.813	0.752	0.639	0.984	<u>0.930</u>	0.884	0.809
CAT-Netv2 [13]	0.680	0.635	0.977	0.803	0.942	0.838	0.667	0.580	0.747	0.525	0.803	0.676
ManTraNet [12]	0.760	0.500	0.810	0.500	0.644	0.500	0.778	0.500	0.874	0.500	0.773	0.500
PSCC-Net [10]	0.657	0.473	0.300	0.604	0.869	0.520	<u>0.777</u>	0.515	0.650	0.458	0.651	0.514
SPAN [15]	0.670	0.235	0.999	0.759	0.480	0.112	0.475	0.298	0.669	0.233	0.659	0.327
CR-CNN [14]	0.553	0.391	0.755	0.631	0.670	0.481	0.589	0.447	0.576	0.289	0.629	0.448
MVSS-Net [9]	0.733	0.514	0.984	0.729	0.932	0.528	0.654	0.117	0.552	0.358	0.771	0.449
Late Fusion (Sec. III-C)	0.792	0.720	0.977	0.822	0.930	0.860	0.760	<u>0.677</u>	0.958	0.830	0.884	0.782
Early Fusion (Sec. III-D)	0.839	0.770	0.996	<u>0.962</u>	0.929	<u>0.845</u>	0.755	0.660	<u>0.966</u>	0.935	0.897	0.834
MMFusion (Early Fusion with FRD)	0.837	0.765	0.998	0.814	0.931	0.860	0.775	0.699	0.923	0.735	<u>0.893</u>	0.776

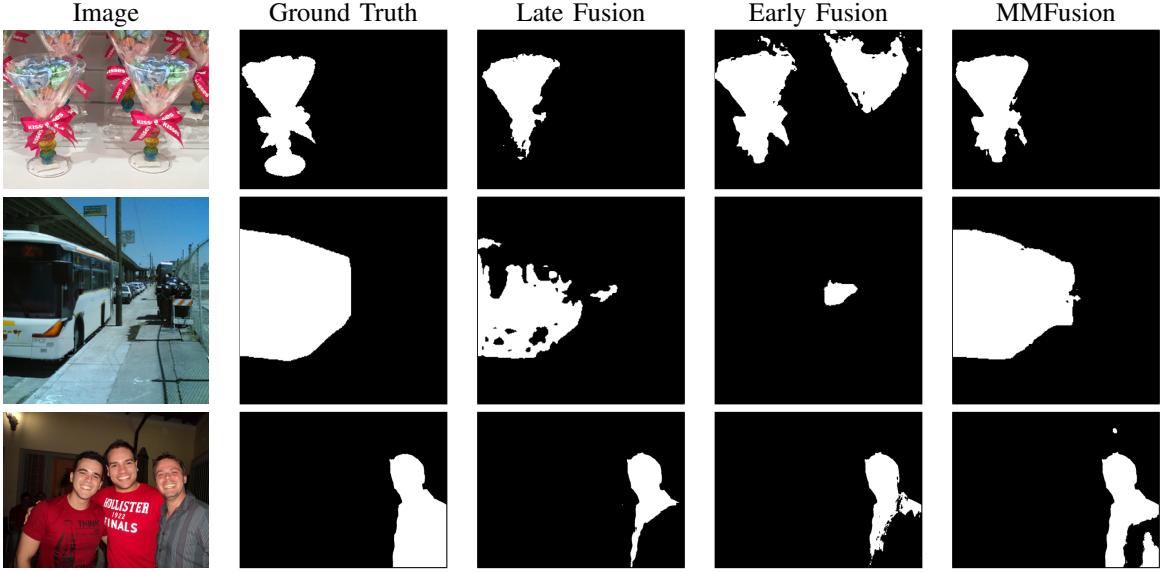


Fig. 6: Qualitative results, showing for each image the Ground Truth mask of the manipulated region and the corresponding prediction of each of the examined / proposed filter fusion approaches. Source dataset of each image: top row: Coverage; middle row: CocoGlide; bottom row: DSO-1.

TABLE VI: Average pixel-level F1 scores for the localization task on the considered ablation study variants and datasets. Parameter count in Millions. Runtime in milliseconds for a single image of the Casiav1+ dataset on an RTX 3090 GPU. Best scores (higher for avg F1, lower for params/runtime) in bold and second best scores underlined.

Version	Coverage	Columbia	Casiav1+	CocoGlide	DSO-1	AVG	Params(M)	Runtime(ms)
CMX (NP++)	0.577	0.884	0.761	0.516	0.895	0.726	68.3	34.9
CMX (Bayar)	0.592	0.872	0.774	0.566	<u>0.776</u>	0.716	68.1	<u>34.2</u>
CMX (SRM)	0.630	0.834	<u>0.791</u>	<u>0.585</u>	0.792	0.726	68.1	34.0
Late Fusion (Sec. III-C) - No weight sharing	0.611	0.912	0.760	0.566	0.785	0.727	200.7	79.1
Late Fusion (Sec. III-C)	0.641	0.864	0.775	0.574	0.899	<u>0.751</u>	152.3	77.2
Early Fusion (Sec. III-D)	<u>0.663</u>	<u>0.888</u>	0.784	0.553	0.863	0.750	68.9	42.0
MMFusion (Early Fusion with FRD)	0.700	0.876	0.794	0.591	0.866	0.765	68.9	43.6

NoisePrint++ are kept frozen. We can see that NoisePrint++’s editing-history-based training helps achieve the best performance on DSO-1, where manipulations are covered using post-processing operations, while SRM and Bayar perform better in CocoGlide and Coverage. Coverage contains only copy-move manipulations for which NoisePrint’s++ camera model identification might not provide robust enough forensic traces, whereas CocoGlide’s manipulations are diffusion-based inpaintings potentially resulting in distinct artifacts that

diverge from conventional editing histories. Consequently, NoisePrint++ encounters difficulties in effectively handling such cases. We also compare all methods that use a single forensic filter to our multi-modal fusion approaches and we can see that the latter effectively combine the forensic traces provided by the different filters, resulting in increased performance. To substantiate our rationale for introducing shared weights between RGB branches in order to enhance regularization within the late fusion paradigm, we additionally

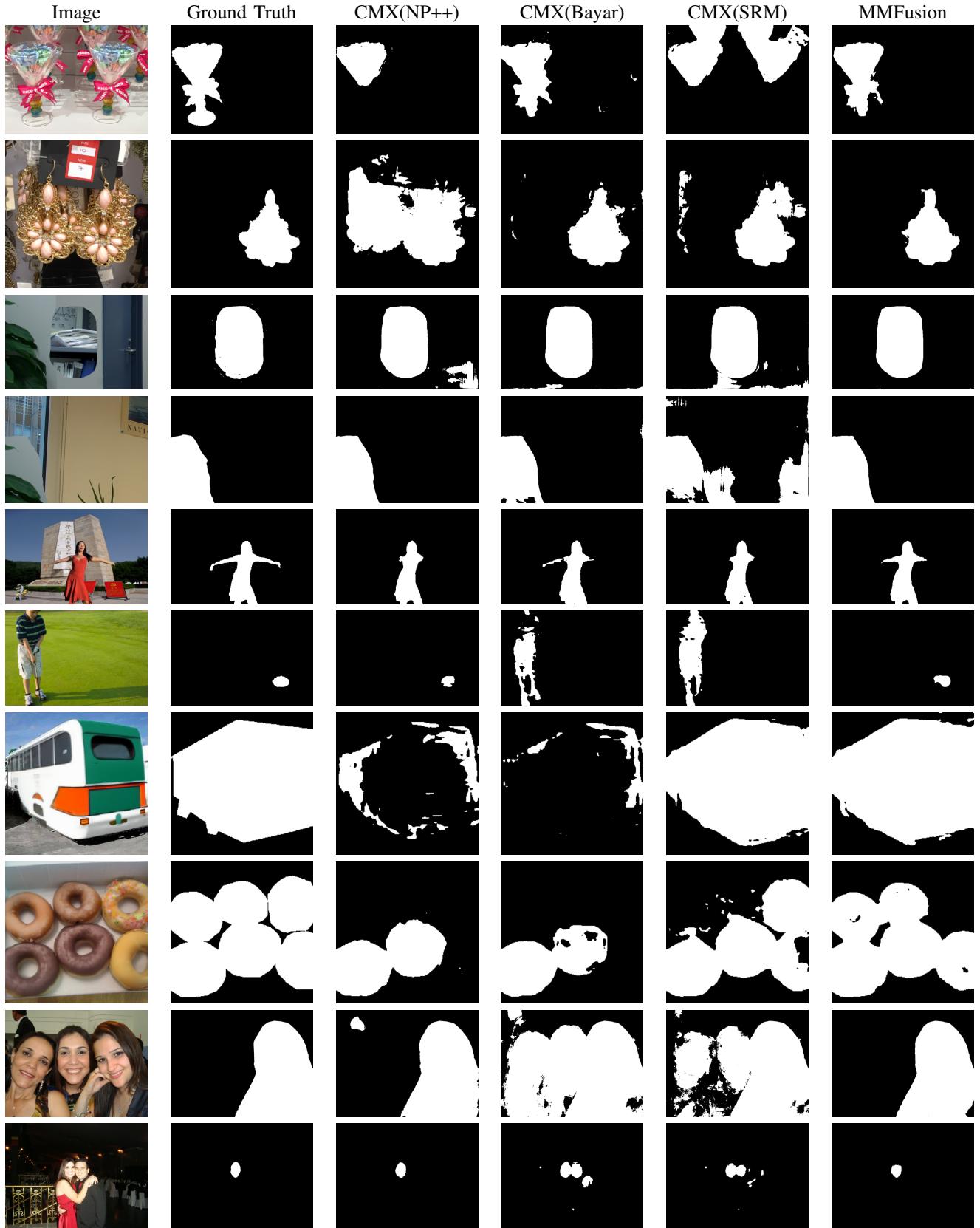


Fig. 7: Qualitative results, showing for each image the Ground Truth mask of the manipulated region and the corresponding predictions of the MMFusion approach as well as of the CMX architecture that uses each filter separately. Source dataset of each image: rows 1-2: Coverage; rows 3-4: Columbia; rows 5-6: Casiav1+; rows 7-8: CocoGlide; rows 9-10: DSO-1.

evaluate a variation of our method that does not employ weight sharing, and we observe that weight-sharing does contribute to improved performance.

We also qualitatively compare our proposed MMFusion approach with the dual-branch CMX architecture that uses each filter separately in Fig 7, and we confirm the ability of MMFusion to effectively combine the information of each filter. More specifically, for the examples from the Coverage dataset (rows 1 and 2), MMFusion accurately localizes the manipulated region that is shown in the ground truth mask, while the SRM and Noiseprint++ models either over-estimate or miss crucial parts. In the splicing samples from the Columbia dataset (rows 3 and 4), the MMFusion-generated localization predictions are the least noisy ones among the results of all compared approaches. On the images of the Casiav1+ dataset (rows 5 and 6), MMFusion correctly identifies the manipulated region of the woman in the red dress, including most of the parts of her arms that other methods missed, and avoiding the incorrect detection of the person playing golf by the Bayar and SRM filters. Similar comments can be made by looking at the CocoGlide images (rows 7 and 8), where, for the bus example Noiseprint++ and Bayar leave gaps in the prediction while in the other image they outright miss detecting some of the donuts as manipulated, contrary to MMFusion that gives the closest localization result to the ground truth. Closing with some samples from the DSO-1 dataset (rows 9 and 10), we observe that Bayar and SRM over-estimate manipulations in the presence of people and faces, with our approach gives the cleanest and most accurate result.

D. Evaluation and Comparisons on Video Manipulation Datasets

We further evaluate our proposed method MMFusion on the VCMS, VPVM, VPIM [40] video datasets, which contain manipulated videos, and compare our approach with the state-of-the-art video manipulation detection model VideoFACT [40] and, by extension, with image manipulation detection methods that were used for comparisons in [40]. As reported in the latter, VideoFACT was trained on a combination of the VCMS, VPVM and VPIM video datasets and three image datasets. For MMFusion, we assess models of it trained on different datasets: a model that is trained on image data only (i.e. the same trained model that is evaluated in the preceding sections), a model that is trained only on the training splits of the employed video datasets (VCMS, VPVM, VPIM), and a model that is trained on all of our training data (i.e. all images and video frames of Table II). The results are presented in Tables VII and VIII.

We observe that even without training on the video datasets, our model reaches state-of-the-art performance on the VCMS dataset that contains splicing manipulations while it outperforms other IMLD methods on the VPVM and VPIM datasets, thus establishing a new baseline of VMLD performance for IMLD models. Our models trained on video-only and both image-video data also achieve state-of-the-art performance across all datasets, showcasing that designing complex temporal modules that are also computationally expensive is not

necessary to achieve state-of-the-art performance on VMLD tasks; simply training on video data should suffice, at least for the types of video manipulations present on our evaluation datasets. This also exposes the need for new more sophisticated VMLD datasets that contain manipulations with more complex temporal elements, in order to be able to more accurately compare VMLD models.

TABLE VII: Average pixel-level F1 scores for the video localization task on the considered models and datasets. Best (higher) scores in bold and second best scores underlined. Results for all models except for the proposed ones are taken from [40].

Model	VCMS	VPVM	VPIM	Avg
VIDEOFACT [40]	0.526	0.697	0.547	0.590
NoisePrint [22]	0.030	0.013	0.010	0.018
MantraNet [12]	0.114	0.145	0.064	0.108
MVSS-Net [9]	0.557	0.279	0.042	0.293
MMFusion (Image Data)	0.838	0.520	0.229	0.529
MMFusion (Video Data)	0.952	0.945	0.686	0.861
MMFusion (All data)	<u>0.921</u>	<u>0.898</u>	<u>0.579</u>	<u>0.799</u>

TABLE VIII: Balanced accuracy scores for the video detection task on the considered models and datasets. Best (higher) scores in bold and second best scores underlined. Results for all models except for the proposed ones are taken from [40].

Model	VCMS	VPVM	VPIM	Avg
VIDEOFACT [40]	0.987	<u>0.950</u>	0.797	<u>0.911</u>
NoisePrint [22]	0.500	0.500	0.500	0.500
MantraNet [12]	0.500	0.500	0.500	0.500
MVSS-Net [9]	0.602	0.529	0.492	0.541
MMFusion (Image Data)	0.915	0.650	0.510	0.692
MMFusion (Video Data)	<u>0.963</u>	0.962	0.878	0.934
MMFusion (All data)	<u>0.923</u>	<u>0.923</u>	<u>0.822</u>	0.889

E. Robustness Analysis

In this section, we include experiments performed on images with varying quality degradations to demonstrate the robustness of our approach, similarly to Guillaro et al [11]. We use the Casiav1+ dataset and perform Gaussian blurring with different kernel sizes (3, 5, 7, 9, 11, 13) and JPEG compression with varying quality factors (100, 90, 80, 70, 60, 50) and compare the emerging pixel-level F1 scores to our baseline, TruFor. The findings depicted in Fig. 8 demonstrate that our MMfusion architecture exhibits good robustness across a broad spectrum of degradations, maintaining a consistent advantage over TruFor.

V. EXPLAINABILITY OF IMLD MODELS: QUANTIFYING THE IMPORTANCE OF DIFFERENT FORENSIC FILTERS

Image manipulation localization and detection as a Machine Learning task is inherently explainable to some extent, as the localization map predicted can serve as the explanation for the detection prediction (e.g. a “manipulated” classification decision can be explained by “the region shown in this localization map is predicted to be manipulated”, in an analogy to the form of explanations produced by e.g. T-TAME [55] for ImageNet classifiers). Despite this, we investigate how

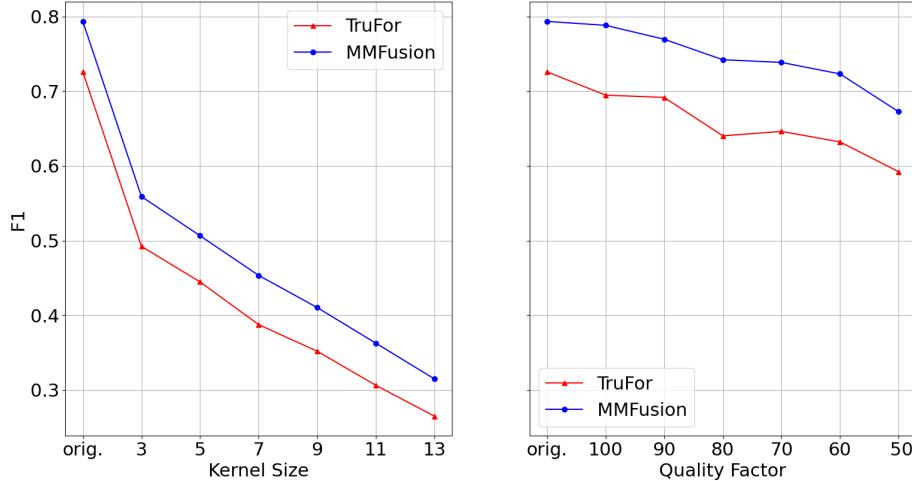


Fig. 8: Robustness analysis with regards to Gaussian blur (left) and JPEG compression (right). Higher F1 values are better.

TABLE IX: Average drop in the pixel-level F1 scores (calculated using the ground truth mask for the localization task), before and after masking each modality with either zeros (above the diving horizontal line) or with another random image (below the diving horizontal line). Best (higher) scores in bold and second best scores underlined.

Masked Modality → Mask Type	Coverage	Columbia	Casiav1+	CocoGlide	DSO-1	AVG
NoisePrint → 0	0.0225	0.0236	0.0168	-0.0222	0.2015	0.0484
Bayar Conv → 0	-0.0026	-0.0051	0.0154	-0.0442	0.0253	-0.0022
SRM → 0	0.2367	0.0276	<u>0.0649</u>	0.1376	0.0709	0.1075
NoisePrint → Random Image	0.0225	0.0155	0.0200	-0.0268	0.2277	0.0487
Bayar Conv → Random Image	0.0267	-0.0041	0.0175	-0.0072	0.0385	0.0143
SRM → Random Image	0.2269	0.0393	<u>0.0677</u>	0.1505	<u>0.1487</u>	0.1266

TABLE X: Drop in Prediction Quality (PQ) measure scores using the output of the unmasked prediction as ground truth mask for the localization task, after masking each modality with zeros (above the diving horizontal line) or with another random image (below the diving horizontal line). Best (lower) scores in bold and second best scores underlined.

Masked Modality → Mask Type	Coverage	Columbia	Casiav1+	CocoGlide	DSO-1	AVG
NoisePrint → 0	0.8215	0.9381	0.8605	0.5689	0.7349	0.7848
Bayar Conv → 0	0.8079	0.9747	<u>0.8291</u>	<u>0.5412</u>	0.9576	0.8221
SRM → 0	0.5521	0.9356	0.7934	0.3636	0.9078	0.7105
NoisePrint → Random Image	0.8340	0.9489	0.8491	0.5889	0.7117	0.7865
Bayar Conv → Random Image	0.7892	0.9726	0.8299	<u>0.5272</u>	0.9317	0.8101
SRM → Random Image	0.5455	0.9231	0.7960	<u>0.3370</u>	0.8051	0.6813

to extend the explainability of our multi-modal models for IMLD, given the high importance for end-users to be offered rich insights about the decision-making processes, to allow for greater transparency and trustworthiness of the classification decisions. This investigation builds on our preceding ablation study, which revealed that different forensic filters exhibit complementary performance characteristics (Sec. IV-C). The importance of each filter becomes apparent when evaluating its effectiveness against distinct types of manipulated images; for instance, some filters are particularly adept at identifying copy-move forgeries seen in the Coverage dataset, while others excel with splicing forgeries from the Columbia dataset, which lacks post-processing. To dive deeper into this, we employ a perturbation-based explanation method, in the spirit of methods like LIME and SHAP [56], [57]: we mask one modality (i.e. filter output) at each time and observe MMFusion’s resulting drop in performance. Essentially, we

replace the input of the chosen filter with either zeros or a random pristine image and quantify the importance of the filter as the drop in localization F1 (higher drop means the filter is more important). This approach highlights the filter’s reliance on specific data features. The results are displayed in Table IX. Our tests across different datasets, each exhibiting different manipulations (as detailed in Sec. IV-A2), demonstrate that each filter is tailored to detect particular forgery characteristics, thereby offering a comprehensive analysis framework for diverse forensic scenarios.

This explanation method, however, relies on our knowledge of the ground truth mask for the image. In order to be able to provide explanations for in-the-wild images, we expand our methodology and propose Drop in Prediction Quality (PQ) as a new evaluation measure. Drop in Prediction Quality is calculated as the F1 measure for the masked prediction using the original unmasked prediction as the ground truth. For the

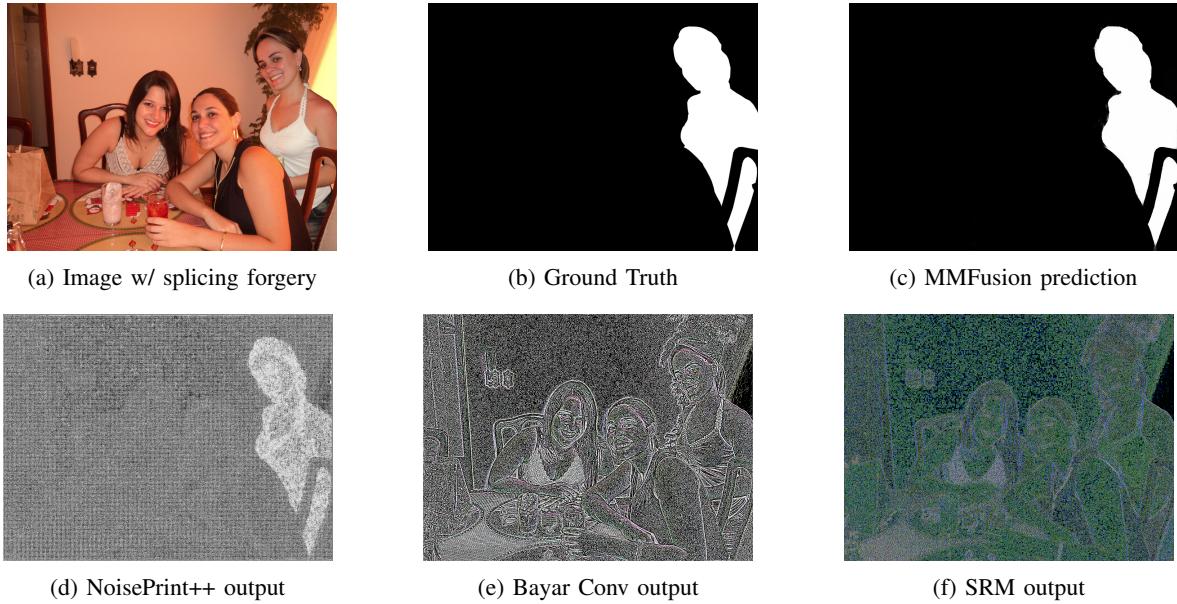


Fig. 9: Visualization of the filter outputs for a DSO-1 dataset image. Top row: (a) Input image with splicing forgery, (b) Ground Truth mask of the manipulated region, (c) Prediction/Detection mask of MMFusion. Bottom row: (d)-(f) Output of each filter.

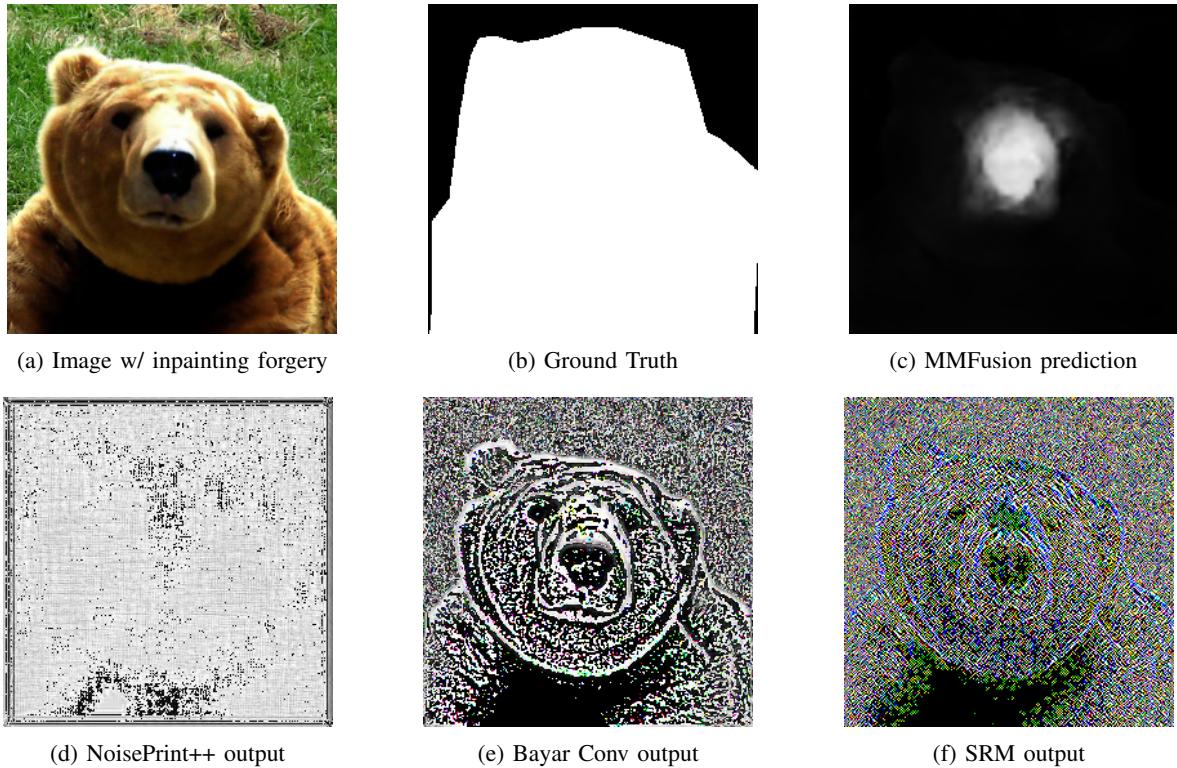


Fig. 10: Visualization of the filter outputs for a CocoGlide dataset image. Top row: (a) Input image with inpainting forgery, (b) Ground Truth mask of the manipulated region, (c) Prediction/Detection mask of MMFusion. Bottom row: (d)-(f) Output of each filter.

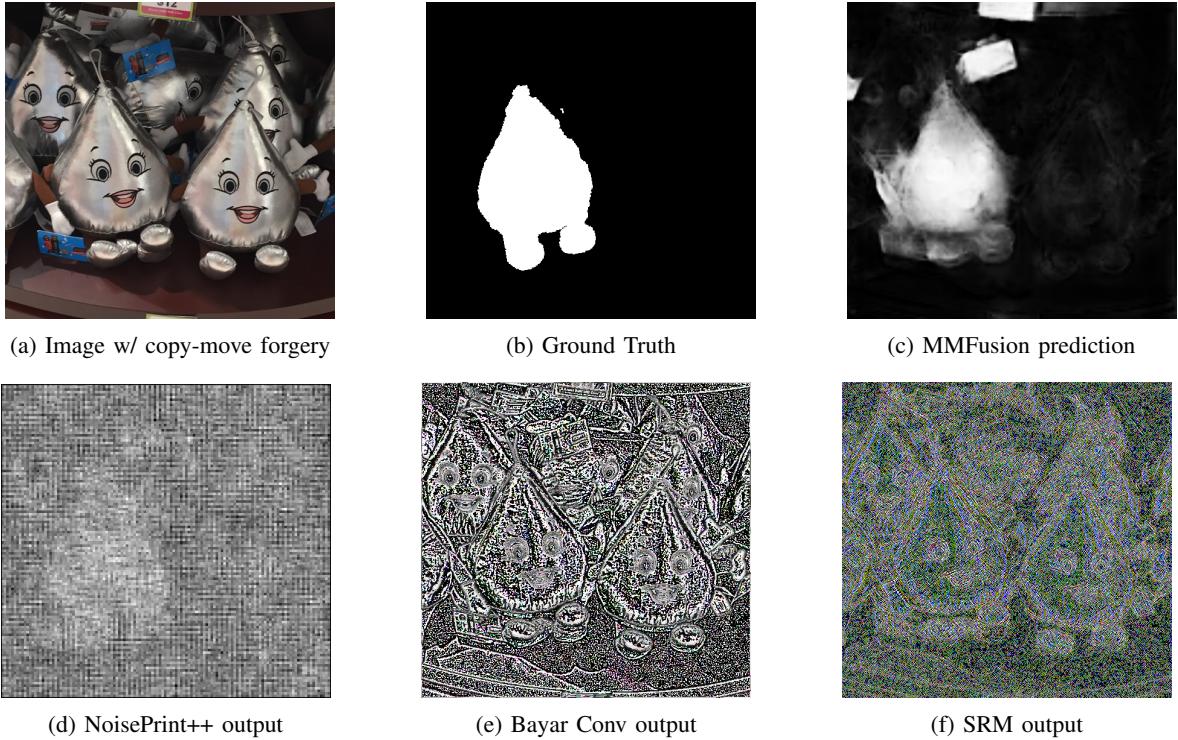


Fig. 11: Visualization of the filter outputs for a Coverage dataset image. Top row: (a) Input image with inpainting forgery, (b) Ground Truth mask of the manipulated region, (c) Prediction/Detection mask of MMFusion. Bottom row: (d)-(f) Output of each filter.

new PQ measure, lower value means the masked modality is more important. We present the blind explanations results in Table X. We observe that these results are consistent with those reported in Table IX in highlighting SRM and NoisePrint++ as the most important overall filters. Specifically, the SRM filter is the most important for copy-move and inpainting forgeries, whereas NoisePrint++ is very helpful for recognizing splicing forgeries. Our hypothesis posits that SRM’s effectiveness stems from its sensitivity to edge artifacts produced during the manipulation process, while NoisePrint++ excels at identifying differences in texture between the source and target image that exist in splicing forgeries. To provide more insight on what the output of each filter looks like, a few examples of filter outputs for images with different manipulations are illustrated in Figures 9, 10 and 11.

VI. CONCLUSION

In this work, we expand an existing encoder-decoder architecture for image manipulation localization and detection (IMLD) to support multiple forensic filters as inputs. We examine two filter fusion paradigms: one that generates independent features from each forensic filter before fusing them (late fusion), and another that performs early mixing of modal outputs to produce combined features (early fusion). By leveraging three forensic filters, i.e. Bayar convolution, SRM and NoisePrint++, we show that these filters provide distinct and complementary forensic capabilities and can be effectively combined, as hypothesized. We then introduce a feature re-weighting decoder and deploy it alongside early fusion to

propose the MMFusion architecture. Extensive experiments demonstrate that MMFusion achieves state-of-the-art performance across multiple image datasets, showcasing good generalization and robustness, and its effectiveness in leveraging diverse forensic artifacts from different filters. Additionally, we apply MMFusion to video manipulation datasets, also reaching state-of-the-art performance. Finally, we further assess the contribution of each forensic filter to the MMFusion model’s decisions.

REFERENCES

- [1] Avisek Lahiri, Arnav Kumar Jain, Sanskar Agrawal, Pabitra Mitra, and Prabin Kumar Biswas. Prior guided gan based semantic inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 13696–13705, 2020.
- [2] Xian Zhang, Xin Wang, Canghong Shi, Zhe Yan, Xiaojie Li, Bin Kong, Siwei Lyu, Bin Zhu, Jiancheng Lv, Youbing Yin, et al. De-gan: Domain embedded gan for high quality face image inpainting. *Pattern Recognition*, 124:108415, 2022.
- [3] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the 39th International Conference on Machine Learning*, pages 16784–16804. PMLR, 2022.
- [4] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22428–22437, 2023.
- [5] Denis Teyssou, Jean-Michel Leung, Evlampios Apostolidis, Konstantinos Apostolidis, Symeon Papadopoulos, Markos Zampoglou, Olga Papadopoulou, and Vasileios Mezaris. The InVID plug-in: web video verification on the browser. In *Proc. Int. Workshop on Multimedia Verification (MuVer) at ACM Multimedia 2017*, pages 23–30, 2017.

- [6] Ronald Salloum, Yuzhuo Ren, and C-C Jay Kuo. Image splicing localization using a multi-task fully convolutional network (mfcn). *Journal of Visual Communication and Image Representation*, 51:201–209, 2018.
- [7] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Copy-move forgery detection based on patchmatch. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5312–5316. IEEE, 2014.
- [8] Haodong Li, Weiqi Luo, and Jiwu Huang. Localization of diffusion-based inpainting in digital images. *IEEE Transactions on Information Forensics and Security*, 12(12):3050–3064, 2017.
- [9] 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 (CVPR)*, pages 14185–14193, 2021.
- [10] 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.
- [11] 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.
- [12] Yue Wu, Wael AbdAlmageed, and Premkumar 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 (CVPR)*, pages 9543–9552, 2019.
- [13] Myung-Joon Kwon, Seung-Hun Nam, In-Jae Yu, Heung-Kyu Lee, and Changick Kim. Learning jpeg compression artifacts for image manipulation detection and localization. *International Journal of Computer Vision*, 130(8):1875–1895, 2022.
- [14] Chao Yang, Huizhou Li, Fangting Lin, Bin Jiang, and Hao Zhao. Constrained r-cnn: A general image manipulation detection model. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020.
- [15] Xuefeng Hu, Zhihan Zhang, Zhenye Jiang, Syomantak Chaudhuri, Zhenheng Yang, and Rami Neftaia. Span: Spatial pyramid attention network for image manipulation localization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pages 312–328. Springer, 2020.
- [16] Junke Wang, Zuxuan Wu, Jingjing Chen, Xintong Han, Abhinav Shrivastava, Ser-Nam Lim, and Yu-Gang Jiang. Objectformer for image manipulation detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2364–2373, 2022.
- [17] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012.
- [18] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1053–1061, 2018.
- [19] Dong Li, Jiaying Zhu, Menglu Wang, Jiawei Liu, Xueyang Fu, and Zheng-Jun Zha. Edge-aware regional message passing controller for image forgery localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8222–8232, 2023.
- [20] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16317–16326, 2021.
- [21] 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*, page 5–10, 2016.
- [22] Davide Cozzolino and Luisa Verdoliva. Noiseprint: A cnn-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security*, 15:144–159, 2019.
- [23] Konstantinos Triaridis and Vasileios Mezaris. Exploring multi-modal fusion for image manipulation detection and localization. In *International Conference on Multimedia Modeling*, pages 198–211. Springer, 2024.
- [24] Alessandro Piva. An overview on image forensics. *International Scholarly Research Notices*, 2013(1):496701, 2013.
- [25] Jessica Fridrich. Digital image forensics. *IEEE Signal Processing Magazine*, 26(2):26–37, 2009.
- [26] Zhigang Fan and Ricardo de Queiroz. Maximum likelihood estimation of jpeg quantization table in the identification of bitmap compression history. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 1, pages 948–951. IEEE, 2000.
- [27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015.
- [28] Zenan Shi, Xuanjing Shen, Hui Kang, and Yingda Lv. Image manipulation detection and localization based on the dual-domain convolutional neural networks. *IEEE Access*, 6:76437–76453, 2018.
- [29] H. Song, Baichuan Lin, and D. Ye. Tri-path backbone network for image manipulation localization. *IEEE Access*, 12:83217–83227, 2024.
- [30] Amir Etefaghi Daryani, Mahdieh Mirmahdi, Ahmad Hass nanopour, Hatef Oetrosi Shahreza, Bian Yang, and Julian Fierrez. Irl-net: Inputted region localization network via spatial attention. *IEEE Access*, 11:115677–115687, 2023.
- [31] Jiaming Zhang, Huayao Liu, Kailun Yang, Xinxin Hu, Ruiping Liu, and Rainer Stiefelhagen. Cmx: Cross-modal fusion for rgb-x semantic segmentation with transformers. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–16, 2023.
- [32] Nitin Arvind Shelke and Singara Singh Kasana. A comprehensive survey on passive techniques for digital video forgery detection. *Multimedia Tools and Applications*, 80(4):6247–6310, 2021.
- [33] Markos Zampoglou, Foteini Markatopoulou, Gregoire Mercier, Despoina Touska, Eylampos Apostolidis, Symeon Papadopoulos, Roger Cozien, Ioannis Patras, Vasileios Mezaris, and Ioannis Kompatiari. Detecting tampered videos with multimedia forensics and deep learning. In *MultiMedia Modeling: 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8–11, 2019, Proceedings, Part I 25*, pages 374–386. Springer, 2019.
- [34] Alessandra Gironi, Marco Fontani, Tiziano Bianchi, Alessandro Piva, and Mauro Barni. A video forensic technique for detecting frame deletion and insertion. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6226–6230. IEEE, 2014.
- [35] Chengjiang Long, Eric Smith, Arslan Basharat, and Anthony Hoogs. A c3d-based convolutional neural network for frame dropping detection in a single video shot. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1898–1906. IEEE, 2017.
- [36] Tamer Shanableh. Detection of frame deletion for digital video forensics. *Digital Investigation*, 10(4):350–360, 2013.
- [37] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8261–8265. IEEE, 2019.
- [38] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. Deepforensics-1.0: A large-scale dataset for real-world face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2889–2898, 2020.
- [39] Zhiyuan Yan, Yong Zhang, Yanbo Fan, and Baoyuan Wu. Ucf: Uncovering common features for generalizable deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22412–22423, 2023.
- [40] Tai D Nguyen, Shengbang Fang, and Matthew C Stamm. Videofact: Detecting video forgeries using attention, scene context, and forensic traces. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8563–8573, 2024.
- [41] Enze Xie, Wenhui Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.
- [42] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.
- [43] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12695–12705, 2020.
- [44] Yunfeng Fan, Wenchao Xu, Haozhao Wang, Junxiao Wang, and Song Guo. Pmr: Prototypical modal rebalance for multimodal learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20029–20038, 2023.

- [45] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems*, 34:30392–30400, 2021.
- [46] Bihang Wen, Ye Zhu, Ramanathan Subramanian, Tian-Tsong Ng, Xuanjing Shen, and Stefan Winkler. Coverage—a novel database for copy-move forgery detection. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 161–165. IEEE, 2016.
- [47] 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.
- [48] James Ze Wang, Jia Li, and Gio Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947–963, 2001.
- [49] Jing Dong, Wei Wang, and Tieniu 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.
- [50] Tiago José De Carvalho, Christian Riess, Elli Angelopoulou, Helio Pedrini, and Anderson de Rezende Rocha. Exposing digital image forgeries by illumination color classification. *IEEE Transactions on Information Forensics and Security*, 8(7):1182–1194, 2013.
- [51] 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.
- [52] Adam Novozamsky, Babak Mahdian, and Stanislav 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 (WACVW)*, pages 71–80, 2020.
- [53] Vladimir V Knyaz, Vladimir Knyaz, and Fabio Remondino. The point where reality meets fantasy: Mixed adversarial generators for image splice detection. *Advances in Neural Information Processing Systems*, 32, 2019.
- [54] Jiaming Zhang, Ruiping Liu, Hao Shi, Kailun Yang, Simon Reiß, Kunyu Peng, Haodong Fu, Kaiwei Wang, and Rainer Stiefelhagen. Delivering arbitrary-modal semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1136–1147, 2023.
- [55] Mariano V. Ntougkas, Nikolaos Gkalelis, and Vasileios Mezaris. T-tame: Trainable attention mechanism for explaining convolutional networks and vision transformers. *IEEE Access*, pages 1–1, 2024.
- [56] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.
- [57] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.