Deep Video Inpainting Detection

Peng Zhou[1]
pengzhou@umd.edu
Ning Yu[12]
ningyu@cs.umd.edu
Zuxuan Wu[1]
zxwu@cs.umd.edu
Larry S. Davis[1]
lsd@umiacs.umd.edu
Abhinav Shrivastava[1]
abhinav@cs.umd.edu
Ser-Nam Lim[3]
sernamlim@fb.com

[1] University of Maryland
College Park, MD, USA
[2] Max Planck Institute for Informatics
Saarbrücken, Saarland, Germany
[3] Facebook AI
New York City, NY, USA

Abstract

Video inpainting has become an increasingly matured forensics technique and has caused visual misinformation on social media. Yet the countermeasures to detect inpainted regions in videos have received little attention, leaving such threats out of control. To pioneer a mitigation solution, we introduce VIDNet, the first study of learning-based video inpainting detection, which contains a two-stream encoder-decoder architecture with attention module. To reveal artifacts encoded in compression, VIDNet additionally takes in Error Level Analysis frames to augment RGB frames, producing multimodal features at different levels with an encoder. Exploring spatial and temporal relationships, these features are further decoded by a Convolutional LSTM to predict masks of inpainted regions. In addition, when detecting whether a pixel is inpainted or not, we present a quad-directional local attention module that borrows information from its surrounding pixels from four directions. Extensive experiments validate the significant advantages of VIDNet over alternative inpainting detection baselines, as well as its generalization on unseen videos. We have released our code in https://github.com/pengzhou1108/VIDNet.

1 Introduction

Fake news [2] and deepfake [1] issues arise recently due to the abuse of advanced image or video manipulation techniques, and the remedy for it has become an increasing necessity. Among those techniques, video inpainting, which completes corrupted or missing regions in a video sequence, has achieved impressive progress over the years [4, 12, 16, 17, 25, 26, 38, 39, 40]. The ability to produce realistic videos that can be used in applications like video restoration, virtual reality, *etc*., while appealing, brings significant security concerns at the

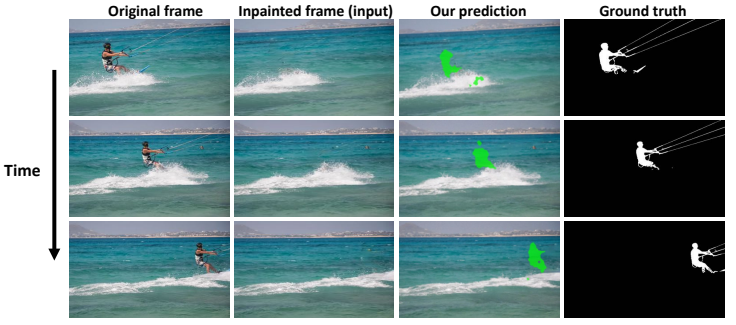

Figure 1: **Problem introduction**. Given an inpainted video (second column), we localize the inpainted region both spatially and temporally.

same time since these techniques can also be used maliciously. By removing objects that could serve as evidence, malicious inpainting can result in serious legal and social implications including swaying a jury, accelerating the spread of misinformation on social platforms, *etc*. To fight against the ubiquitous fake inpainted videos, our goal in this work is to develop a framework for detecting inpainted videos constructed with state-of-the-art methods (see Fig. 1 for a conceptual overview).

Although there are recent studies on detecting tampered regions in images [7, 13, 37, 48], very limited effort has been devoted to video inpainting detection. For image-based manipulation detection, existing approaches either focus on spliced regions or "deepfake"-style face replacement instead of object removal based on inpainting. Additionally, most of them are designed specifically for images [13, 36] only and suffer from poor performance on videos. Learning robust video representations can help mitigate issues with single image detection.

In light of this, we introduce VIDNet, a video inpainting detection network, which is an encoder-decoder architecture with a quad-directional local attention module to predict inpainted regions in videos (as is shown in Fig. 2). In particular, at each time step, VIDNet's encoder takes as input the current RGB frame, truncated from a pretrained VGG network [30]. Since videos are compressed based on discrete cosine transforms (DCT) and frames extracted are usually stored in JPEG format, we leverage ELA [15] images as an additional input to the encoder to reveal artifacts like compression inconsistency (as is shown in Fig. 3). We extract features from both ELA and RGB images with the encoder, producing five different multimodal features at different scales, that are further used jointly to train our inpainting detector. In addition, given a missing region to fill in, inpainting methods leverage information from surrounding pixels of the region to make the region coherent spatially. Motivated by this, for RGB features from the last layer of the encoder, we introduce a quad-directional local attention module to attend to the neighbors of a pixel, allowing us to explicitly model spatial dependencies among different pixels during detection.

Finally, with multimodal features encoded at different scales, we leverage a four-layer Convolutional LSTM, serving as a decoder for inpainting detection. More specifically, the ConvLSTM at a certain layer not only takes in features from a previous time step but also features upsampled from a coarse level (*e.g*., a lower decoding layer). In this way both spatial relationships across different scales and temporal dynamics over time are leveraged to produce inpainted masks over time. The framework is trained end-to-end with backpropagation. We conduct experiments on the DAVIS 2016 [27] Dataset and the Free-form Video Inpainting Dataset [4]. VIDNet successfully detects inpainted regions under all different set-

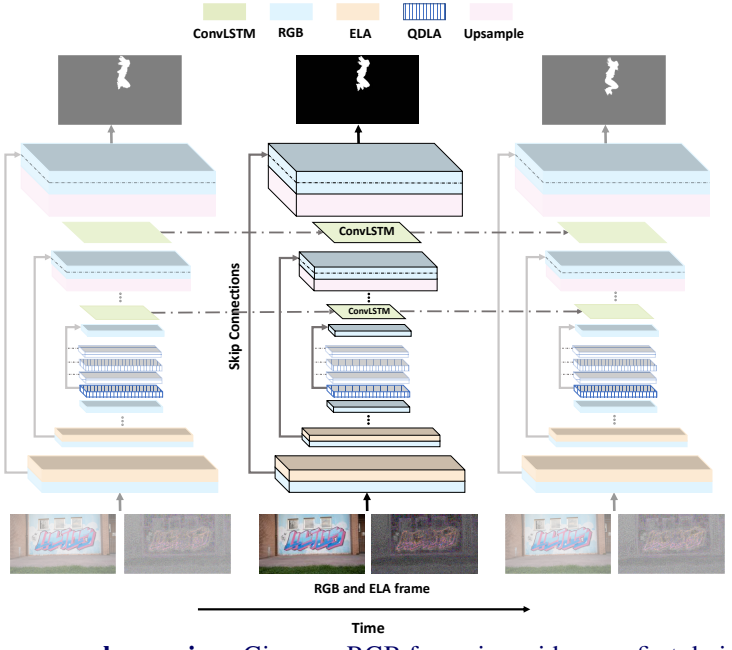

Figure 2: **Framework overview**. Given an RGB frame in a video, we first derive its corresponding ELA frame and compute multimodal features at different scales. We also introduce a quad-directional local attention module (striped) to the last encoded RGB features (colored blue) to explore spatial relationships from four directions. These encoded features are further input into a multi-layer ConvLSTM (colored green) to produce masks of inpainted regions.

tings and outperforms by clear margins competing methods. We also show that VIDNet can be generalized to detect out-of-domain inpainted videos that are unseen during training.

Our contributions are summarized as follows: 1) To the best of our knowledge, this is the first work to formalize the problem of learning-based video inpainting detection. 2) We present an end-to-end framework for video inpainting detection, which leverages multimodal features and models spatial and temporal relationships in videos. 3) We introduce a quad-directional local attention module to explicitly determine if a pixel is inpainted or not by attending to its neighbours.

2 Related Work

Video Inpainting. With the advance of recent image inpainting approaches [10, 11, 14, 19, 21, 26, 38, 40, 47], more recent studies have investigated video inpainting. There are two lines of work — patch based and learning based approaches. For patch based approaches, PatchMatch [3] is a prominent approach which searches for similar patches in the surrounding region iteratively to complete the inpainted region. To achieve better quality, Huang *et al*. [12] explore an optimization based method to match patches and utilize information including color and flow as regularization. On the other hand, learning based approaches have been explored recently. Wang [31] propose a 3D encoder-decoder structure for video inpaining. Afterwards, Xu *et al*. [39] leverages optical flow information to guide inpainting in videos in both forward and backward passes. Similarly, Kim *et al*. [16] estimate the proceed-

ing flow as additional constraint while completing the missing regions. To maintain more frame pixels, Oh *et al*. [25] use gated convolution to inpaint video frames gradually from the reference frame. Lee *et al*. [17] copy and paste future frames to complete missing details in the current frame. In contrast, our approach detects regions inpainted by these approaches.

Manipulation Detection. There are also approaches focusing on manipulation detection. Most of them mainly tackle deep generation [24, 32, 41, 42, 43] or splicing based manipulation [6, 7, 8, 33, 34, 44] and use clues specific to them. In particular, Zhou *et al*. [48] use both RGB and local noise to detect potential regions. Salloum *et al*. [29] rely on boundary artifacts to reveal manipulated regions in a multi-task learning fashion and Zhou *et al*. [49] improve its generalization ability with a generative model. Huh *et al*. [13] use metadata to find inconsistent patches and Wu *et al*. [37] treat it as anomaly detection to learn features in a self-supervised manner. Note that video inpainting is different from splicing or copy-move manipulation techniques. Its goal is to achieve perceptual consistency with the background, where these anomaly detection based approaches would probably fail.

More related to our work are methods for image inpainting detection. [36] is a classical approach that searches for similar patches matched by zero-connectivity. However, high false alarm rates limit their applications in real scenarios. More recently, Zhu *et al*. [50] use CNNs to localize inpainting patches within images. Li *et al*. [18] explore High Pass Filtering (HPF) as the initialization of CNNs for the purpose of distinguishing high frequency noise of natural images from inpainted ones. However, the generalization and robustness is limited as these HPFs are learned given specific inpainting methods. In contrast, we combine both RGB information and ELA features as inputs to VIDNet, and show that our approach generalizes to different inpainting methods. Also, detecting video inpainting requires modeling temporal information which is obviously not considered by image inpainting detection methods.

3 Approach

VIDNet, Video Inpainting Detection Network, is an encoder-decoder architecture (See Fig. 2 for an overview the framework) operating on multimodal features to detect inpainted regions. In addition to RGB video frames, VIDNet utilizes Error Level Analysis frames (Sec. 3.1) to identify artifacts incurred during the inpainting process. Motivated by the fact that inpainting methods typically borrow information from neighbouring pixels of the region to be inpainted, we introduce a multi-head local attention module (Sec. 3.2) which uses adjacent pixels to discover inpainting traces. Finally, we model the temporal relations among different frames with a ConvLSTM (Sec. 3.3). In the following, we describe the components of the model.

3.1 Multimodal Features

Learning a mapping directly from an inpainted RGB frame to a mask that encloses the removed object is challenging, since the RGB space is intentionally modified by replacing regions with their surrounding pixels to appear realistic. To mitigate this issue, we additionally augment RGB information with error level analysis features [35] that are designed to reveal regions with inconsistent compression artifacts in compressed JPEG images. Note although videos are usually compressed in MPEG formats, extracted frames are often times stored in the format of JPEG. More formally, an ELA image is defined as:

$$\mathbf{I}^{\mathrm{ELA}} = |\mathbf{I} - \mathbf{I}^{\mathrm{JPG}}|, \tag{1}$$

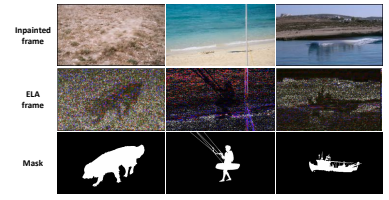

Figure 3: **ELA frame example**. From the top to the bottom: the inpainted RGB frame, its corresponding ELA frame, and the ground-truth inpainting mask. The inpainting artifacts, *e.g.*, the dog, person and ship, stand out in ELA space while not easily seen in the RGB space.

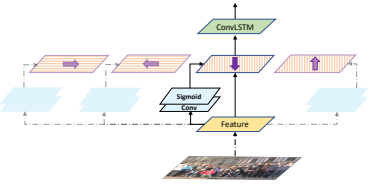

Figure 4: **The quad-directional local attention module**. Given RGB features from the last layer of the encoder, we derive attention maps with convolution and sigmoid layers. To detect whether a pixel is inpainted or not, the module attends to its neighbors from four directions.

where $\mathbf{I}^{\text{ELA}}$ is the ELA image, $\mathbf{I}$ denotes the original image and $\mathbf{I}^{\text{JPG}}$ denotes the recompressed JPEG image from the original image. $|\cdot|$ denotes absolute value.

Fig. 3 illustrates the corresponding ELA images of sampled inpainted frames. Although ELA images have been used in forensics applications [44, 45], they tend to create false alarms when other artifacts like *e.g.*, sharp boundaries, are present in the images, which requires ad-hoc judgement to determine whether a region is tampered. So, instead of only using ELA frames, we augment them with RGB frames as inputs to our encoder (See results in Sec. 4).

In particular, both the RGB and ELA frames are input to a two-stream encoder. Each stream, based on a VGG encoder, transforms the input image to high-level representations with five layers, yielding 5 feature representations at different scales. At each scale, we normalize the corresponding RGB and ELA features, respectively with ℓ_2 normalization, and then apply one convolutional layer to absorb both features into a unified representation:

$$\mathbf{f}_l = \sigma(F([\,\mathbf{f}_l^{\text{RGB}} \mid \mathbf{f}_l^{\text{ELA}}\,])) \quad (l < 5), \tag{2}$$

where $[|]$ denotes feature concatenation, $\mathbf{f}_l$ denotes the feature at l-th layer. $\mathbf{f}_l^{\text{RGB}}$, $\mathbf{f}_l^{\text{ELA}}$ denote the ℓ_2 normalized RGB and ELA features at layer l, respectively. F represents the convolutional layer and σ denotes the activation function. The fused representation at each level is further used for decoding. For $l = 5$, we simply use RGB features as we find that high-level ELA features are not helpful.

3.2 Quad-Directional Local Attention

Inpainting methods aim to replace a region with pixels from its surrounding areas for photorealistic visual effect. Therefore, when determining whether a pixel is inpainted or not, it is important to examine its surrounding pixels. Inspired by recursive filtering techniques that model pixel relations from four directions for edge-preserving smoothing, we introduce a quad-directional local attention module to explore spatial relations among adjacent pixels. Rather than preserve region smoothness and coherence, we design our module to attend to the inconsistency in adjacent features and regions, and thus reveal inpainting artifacts.

As shown in Fig. 4, we learn four attention maps for four directions, left-to-right, right-to-left, top-to-bottom, bottom-to-top, to determine how much information to leverage from the pixels in the corresponding direction based on each map. More specifically, we use $F_{\rightarrow}$,

$F_{\leftarrow}$, $F_{\uparrow}$ and $F_{\downarrow}$ to denote functions that derive attention maps for the left-to-right, right-to-left, top-to-bottom and bottom-to-top four directions. In the following, we consider the left-to-right direction for simplicity. Given features $\mathbf{f}_5$ from the last layer of the RGB stream, we first transform the features with $F_{\rightarrow}$ to have the same dimension as $\mathbf{f}_5$, and then compute an attention map $A_{\rightarrow}$:

$$A_{\rightarrow} = \sigma(F_{\rightarrow}(\mathbf{f}_5; \mathbf{W}_{\rightarrow})), \tag{3}$$

where $\mathbf{W}_{\rightarrow}$ denotes the weights for the convolutional kernel, and σ is the sigmoid function to ensure the attentional weights at each pixel are in the range of $[0,1]$. Then, for each pixel in the feature map, we obtain information from the surrounding pixels as:

$$\mathbf{f}_{5\rightarrow}[k] = (1 - A_{\rightarrow}[k])\mathbf{f}_5[k] + A_{\rightarrow}[k]\mathbf{f}_5[k-1], \tag{4}$$

where k denotes the location of the pixel. Since we are considering attention from the left-to-right direction, $k-1$ indicates the pixel to the left of k. The current value of pixel k is updated with information from its neighboring pixel, the contribution of which is based on the attention map $A_{\rightarrow}$. As a result, we attend to a small local region to compute the refined representation. We can derive $\mathbf{f}_{5\leftarrow}$, $\mathbf{f}_{5\uparrow}$ and $\mathbf{f}_{5\downarrow}$ similarly, and thus we have four different refined representations.

Note that the quad-directional attention module is similar in spirit to recursive filtering. However, in standard recursive filtering, a weight matrix, in the form of an edge map [5] or a weighted map [20], is used as the map A to guide the filtering to restore images or smooth feature maps. In contrast, our filtering can be considered as a form of self-attention—we derive attention maps by modeling similarities in a local region with convolutions conditioned on input features and the resulting maps are in turn used to refine features, allowing pixels to borrow information by attending to its adjacent pixels. In addition, the motivation of our approach can be seen as the "reverse" process of recursive filtering—in recursive filtering, information from surrounding pixels is diffused to make local regions coherent, whereas we wish to detect inconsistent pixels by attending to a neighboring region.

Furthermore, we compute four refined directional feature maps in a parallel way conditioned on the same feature map. An alternative is to generate a single feature representation by sequentially performing attention in four directions, *i.e.*, $\mathbf{f}_{5\rightarrow}$ is used as inputs to generate $\mathbf{f}_{5\leftarrow}$, and so on and so forth, as in [5]. However, we find in Sec. 4 that the parallel multi-head approach offers better results, possibly due to the disentanglement of different directions.

3.3 ConvLSTM Decoder

Temporal information like inconsistency in the inpainted region over time is an important cue for video inpainting detection. To explore temporal relationships among adjacent frames, we use multiple ConvLSTM decoding layers to take features from the encoders and produce predicted detection results, which enables message passing from previous frames. (See Fig. 2) More specifically, the decoder contains four ConvLSTM layers to process features from different spatial scales. At each time step, taking into account both spatial and temporal information, we concatenate the skipped connected feature of the current frame and the upsampled feature from a lower level, as the inputs to the current ConvLSTM layer. Formally, for the t-th time step, the i-th ($2 <= i <= 4$) ConvLSTM computes the hidden states and cell contents for the $t+1$-th time step as:

$$\mathbf{h}_i^{t+1}, \mathbf{c}_i^{t+1} = \mathsf{ConvLSTM}_i(\mathbf{g}_i^t, \mathbf{h}_i^t, \mathbf{c}_i^t), \tag{5}$$

$$\mathbf{g}_i^t = [\, U(\mathbf{h}_{i-1}^t) \,|\, \mathbf{f}_{6-i}^t \,], \tag{6}$$

Methods	VI* IoU/F1	OP* IoU/F1	CP IoU/F1	VI IoU/F1	OP* IoU/F1	CP* IoU/F1	VI* IoU/F1	OP IoU/F1	CP* IoU/F1
NOI [23]	0.08/0.14	0.09/0.14	0.07/ 0.13	0.08/0.14	0.09/0.14	0.07/0.13	0.08/0.14	0.09/0.14	0.07/0.13
CFA [8]	0.10/0.14	0.08/0.14	0.08/0.12	0.10/0.14	0.08/0.14	0.08/0.12	0.10/0.14	0.08/0.14	0.08/0.12
COSNet [27]	0.40/0.48	0.31/0.38	0.36/0.45	0.28/0.37	0.27/0.35	0.38/0.46	0.46/0.55	0.14/0.26	0.44/0.53
HPF [33]	0.46/0.57	0.49/0.62	0.46/0.58	0.34/0.44	0.41/0.51	0.68/0.77	0.55/0.67	0.19/0.29	0.69/0.80
HPF [33] + LSTM	0.50/0.61	0.39/0.51	0.52/0.63	0.26/0.36	0.38/0.44	0.68/0.78	0.53/0.64	0.20/0.30	0.70/0.81
GSR-Net [43]	0.57/0.68	0.50/0.63	0.51/0.63	0.30/0.43	0.74/0.80	0.80/0.85	0.59/0.70	0.22/0.33	0.70/0.77
GSR-Net [43] + LSTM	0.55/0.67	0.51/0.64	0.53/0.64	0.33/0.45	0.60/0.72	0.74/0.83	0.58/0.70	0.21/0.32	0.71/0.81
Ours RGB (baseline)	0.55/0.67	0.46/0.58	0.49/0.63	0.31/0.42	0.71/0.77	0.78/0.86	0.58/0.69	0.20/0.31	0.70/0.82
VIDNet-BN (ours)	**0.62/0.73**	**0.75/0.83**	**0.67/0.78**	0.30/0.42	**0.80/0.86**	**0.84/0.92**	0.58/0.70	0.23/0.32	0.75/0.85
VIDNet-IN (ours)	0.59/0.70	0.59/0.71	0.57/0.69	**0.39/0.49**	0.74/0.82	0.81/0.87	**0.59/0.71**	**0.25/0.34**	**0.76/0.85**

Table 1: **Mean *IoU* and F_1 score comparison on inpainted DAVIS.** The model is trained on VI and OP inpainting, OP and CP inpainting, and VI and CP inpainting respectively (denoted as '*').

where $\mathbf{h}_i^t$ and $\mathbf{c}_i^t$ denote the hidden states and cell states for the i-th ConvLSTM, respectively, and U denotes the function for bilinearly upsampling, which maps the outputs from a lower-level ConvLSTM with smaller feature maps to have the same dimension as the current one. In addition, $\mathbf{f}_{6-i}^t$ is the skip connected feature of the frame t from the encoder.

When $i=1$, $\mathsf{ConvLSTM}_1$takes features from the last layer of the encoder as inputs. Recall that we obtain four refined features based on $\mathbf{f}_5$ with our quad-directional local attention module to identify pixels that are inconsistent with its neighbours from four directions. Thus, we choose one of the four features (e.g., $\mathbf{f}_{5\rightarrow}$) as inputs to the $\mathsf{ConvLSTM}_1$ layer. We then follow the order of $\mathbf{f}_{5\rightarrow}$, $\mathbf{f}_{5\leftarrow}$, $\mathbf{f}_{5\uparrow}$ and $\mathbf{f}_{5\downarrow}$ to obtain all the four directional features.

At each time step, we compute $\mathbf{g}_5^t$ with Eqn. 6 to produce a prediction $\mathbf{p}^t$ for each QDLA direction via one convolutional layer. Finally, to explore non-linear relations among these four directional outputs, we fuse them with one additional convolutional layer to form the final prediction.

During training, we divide each video into N clips with equal clip length. To encourage more intersection with the binary ground truth mask, we use IoU score [28] as our loss function which is formulated as:

$$L(\mathbf{p},\mathbf{y}) = 1 - \frac{\sum \mathbf{p} \odot \mathbf{y}}{\sum(\mathbf{p}+\mathbf{y}-\mathbf{p}\odot\mathbf{y})+\varepsilon}, \tag{7}$$

where $\mathbf{p}$ and $\mathbf{y}$ denote the prediction and the binary ground truth mask, respectively. ε denotes a small number to avoid zero division and $\odot$ represents element-wise multiplication. The loss is updated once the ConvLSTM decoder goes through a single video clip to collect temporal information. By exploring spatial and temporal information recurrently, predictions of inpainted regions become more accurate.

4 Experiment

We compare VIDNet with approaches on manipulation/image inpainting detection in this section to show the advantages of our approach on video inpainting detection both quantitatively and qualitatively.

Dataset and Evaluation Metrics. Since DAVIS 2016 [27] is the most common benchmark for video inpainting, which consists of 30 videos for training and 20 videos for testing, we evaluate our approach on it for inpainting detection. We generate inpainted videos using SOTA video inpainting approaches — VI [16], OP [25] and CP [17], with the ground truth object mask as reference. To show both the performance and generalization, we choose two out of the three inpainted DAVIS for training and testing, leaving one for additional testing.

Methods	VI*	OP*	CP
HPF [18]	0.718	0.640	0.845
GSR-Net [49]	0.762	0.758	0.834
Ours RGB (baseline)	0.722	0.688	0.721
Ours ELA	0.711	0.733	0.850
VIDNet-IN (ours)	**0.778**	**0.768**	**0.884**

Table 2: **Sanity check for inpainting classification AUC comparison.** The results are tested on the three inpainting algorithms, and all the model are trained on VI and OP inpainted DAVIS.

Methods	FVI IoU / F1
NOI [23]	0.062 / 0.107
CFA [8]	0.073 / 0.122
HPF [18]	0.205 / 0.285
HPF [18]+ LSTM	0.177 / 0.271
GSR-Net [49]	0.195 / 0.288
GSR-Net [49] + LSTM	0.188 / 0.281
Ours RGB (baseline)	0.156 / 0.223
VIDNet-IN (ours)	**0.257 / 0.367**

Table 3: **Mean IoU and F1 score comparison on FVI dataset.** All models are trained on VI and OP inpainted DAVIS.

The training/testing split follows DAVIS default setting. We report the F_1 score and mean Intersection of Union (IoU) to the ground truth mask as evaluation metrics.

We compare our method with both video segmentation methods such as COSNet [22] and manipulation detection methods including *NOI* [23], *CFA* [8], *HPF* [18], *GSR-Net* [49] and LSTM version of HPF [18] and GSR-Net [49].

Ours RGB (baseline): Our baseline approach that feeds as input RGB frame only. No QDLA module is applied.

VIDNet-BN (ours): Our batch normalization [15] version.

VIDNet-IN (ours): We report this as our main results, which replaces the batch normalization in encoder by instance normalization.

Sanity Check. Following [13], we first check the ability of our learned model to distinguish between original and inpainted video frames. We compare models trained on VI and OP for simplicity. We add the original uninpainted videos to test sets for evaluation, and average the prediction score for every frame as frame-level score. Afterwards, we report the AUC classification performance in Tab. 2. (Inpainted frames are labeled positive) The comparison between Ours RGB and Ours ELA shows that the ELA features provide more useful clues than the RGB features to distinguish inpainted frames from original ones. With ELA and QDLA features, our model achieves better performance than other methods for all the three inpainting algorithms, validating the advantages of our learned features to classify between inpainted and original videos.

Main Results. Tab. 1 highlights our advantages over other methods. Video segmentation method COSNet captures the flow difference between adjacent frames to segment objects. In contrast, manipulation detection methods are learned to find tamper artifacts and thus yields better performance. Note that without carefully modeling temporal information and consider multiple modalities, adding LSTM doesn't benefit image-based methods like HPF and GSR-Net a lot. In contrast, our approach is designed to detect both spatial and temporal artifacts in two modalities, and thus achieves better performance. Additionally, we show clear improvement over our baseline, indicating the effectiveness of our proposed ELA feature and QDLA module. Comparing across different inpainting algorithms, the performance degrades on the untrained algorithms, indicating a domain shift between trained and untrained inpainting algorithms. However, benefiting from diverse features and more focus on proximity regions, our method still results in better generalization compared with other approaches. Finally, the results indicate that our BN version generally has better performance on the in-domain training inpainting algorithms while IN version shows better generalization on the cross-domain one. Therefore, we provide both results as a trade off.

Methods	VI* IoU / F1	OP* IoU / F1	CP IoU / F1
Ours ELA	0.460 / 0.578	0.509 / 0.631	0.417 / 0.546
Ours RGB (baseline)	0.552 / 0.671	0.456 / 0.580	0.493 / 0.625
Ours w/o QDLA	0.559 / 0.682	0.557 / 0.681	0.512 / 0.644
Ours frame-by-frame	0.558 / 0.683	0.566 / 0.688	0.532 / 0.664
VIDNet-IN (ours)	0.585 / 0.704	0.588 / 0.707	0.565 / 0.685

Table 4: **Ablation analysis.** The model is trained on VI and OP inpainting algorithms (denoted as '*').

Ablation Analysis. We analyze the importance of each key component in our framework and the details are as follows:

Ours ELA: The baseline architecture which only feeds ELA frame as input.

Ours w/o QDLA: Our model that inputs both RGB and ELA frames. No QDLA module is applied.

Ours frame-by-frame: Instead of training with video clip length of 3, we train our full model frame-by-frame.

Tab. 4 displays the comparison results. Compared to baseline, ELA features alone achieve worse performance, which is different from the results in Tab. 2. This is because ELA frames also contain other artifacts like sharp boundaries even though it reveals informative inpainting clues. In contrast, RGB feature is better to capture the boundary details and thus yields more accurate localization. Not surprisingly, with both RGB and ELA features, the network is able to learn more inpainting clues and perform better. Adding the QDLA module introduces feature adjacency relationships and thus leads to further improvement. In addition, the comparison between *Ours frame-by-frame* and our final model verifies the importance of temporal information in video inpainting detection.

Results on Free-form Video Inpainting Dataset. To further test the performance on different dataset, additional evaluation is provided on Free-form Video Inpainting dataset (FVI). FVI dataset [4] provides 100 test videos, which mostly targets multi-instance object removal. We directly apply their approach, which leverages 3D gated convolution encoder-decoder architecture for video inpainting, to generate the 100 inpainted videos. We directly test the models trained on VI and OP inpainted DAVIS.

Tab. 3 displays the comparison results. Since both the dataset and inpainting approach are different, the performance degrades due to the domain shift. However, compared to other approaches, our method still achieves better generalization by a large margin. Also, compared with our baseline model which only uses RGB features, our approach shows clear improvement. This further validates the effectiveness to combine both RGB and ELA features and introduce spatial and temporal information.

Generalization to Different Inpainting Types. To test the model generalization and dependency on inpainting methods, we directly test our model trained on VI and OP on more SOTA flow-based [9] and transformer-based [46] inpainting methods. The results are shown in Tab. 5. Due to the different inpainting mechanisms, we can observe a performance degradation there. However, our model still shows better generalization.

Analysis on Inpainting Ratio. We divide the test samples into 5 groups according to the ratio between inpainting area and the image size and compare the average F1 score in each group in Tab. 6. Performances are worse in both large and small inpainting areas. Small inpainting areas are difficult to predict because the compression inconsistency and inpainting artifacts are hard to detect. For large areas, the model might predicts partial region and lead

Methods	FGVC [9] IoU / F1	STTN [44] IoU / F1
HPF + LSTM	0.10 / 0.154	0.14 / 0.184
GSR-Net + LSTM	0.170 / 0.261	0.383 / 0.509
Ours	**0.201 / 0.301**	**0.410 / 0.540**

Table 5: **Mean IoU and F1 score comparison on DAVIS dataset with different inpainting algorithms.** All models are trained on VI and OP inpainted DAVIS.

Inpainting ratio	VI*	OP*	CP
0-0.1	0.666	0.664	0.596
0.1-0.2	0.831	0.848	0.775
0.2-0.3	0.675	0.763	0.793
0.3-0.4	0.383	0.537	0.540
0.4-0.5	0.154	0.242	0.262

Table 6: **Analysis on inpainting ratio.** The average pixel-level evaluation F1 score under different inpainting ratios. The model is trained on VI and OP inpainted DAVIS. (denoted as '*')

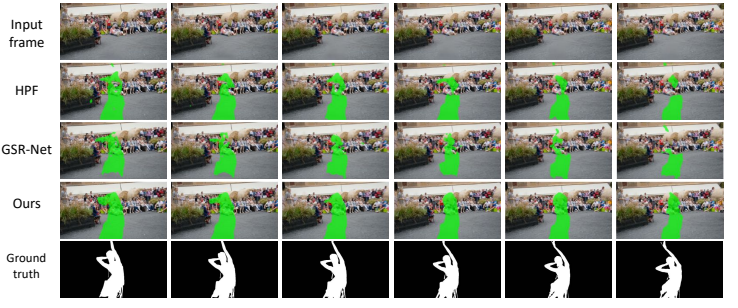

Figure 5: Qualitative visualization on DAVIS. The first row shows the inpainted video frame. The second to fourth row indicates the final predictions from different methods. The fifth row is the ground truth.

to low F1 score; Also, there are limited training samples (less than 10%) for large inpainting ratio and better performance is expected with more largely inpained data.

Qualitative Results. Fig. 5 illustrates the visualization of our predictions versus others under the same setting. Visually, it is clear that our approach is able to obtain a closer prediction to the ground truth than other methods. Specifically, HPF only transfers RGB into noise domain, making it easier to produce false alarm. GSR-Net makes decision based on boundary artifacts, which is not always the strongest artifacts generated by inpainting. In contrast, our prediction utilizes multimodel features and QDLA module, our prediction maintains spatial consistency.

5 Conclusions and Application

We introduce learning based video inpainting detection in this paper. To reveal more inpainting artifacts from different domains, we propose to concatenate both RGB and ELA features. Additionally, we encourage learning from adjacent features in a self-attended manner by introducing a QDLA module. With both the adjacent spatial and temporal information, we make the final prediction through a ConvLSTM based decoder. Our experiments validate the effectiveness of our approach both in-domain and cross-domain. VID methods could be applied as a forensic tool (e.g, in Photoshop) to reveal the fake videos on the internet. Thus, we believe that detecting video inpainting is an important direction for future research.

Acknowledgement We gratefully acknowledge support from Facebook AI and the DARPA MediFor program under cooperative agreement FA87501620191, "Physical and Semantic Integrity Measures for Media Forensics".

References

[1] Deepfake. https://www.deepfakes.club/openfaceswap-deepfakes-software/.

[2] In the age of a.i., is seeing still believing? https://www.newyorker.com/magazine/2018/11/12/in-the-age-of-ai-is-seeing-still-believing.

[3] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *ToG*, 2009.

[4] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-form video inpainting with 3d gated convolution and temporal patchgan. *ICCV*, 2019.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. In *TPAMI*, 2018.

[6] Davide Cozzolino and Luisa Verdoliva. Single-image splicing localization through autoencoder-based anomaly detection. In *WIFS*, 2016.

[7] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv preprint arXiv:1812.02510*, 2018.

[8] Pasquale Ferrara, Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Image forgery localization via fine-grained analysis of cfa artifacts. In *TIFS*, 2012.

[9] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf. Flow-edge guided video completion. In *ECCV*, 2020.

[10] James Hays and Alexei A Efros. Scene completion using millions of photographs. *TOG*, 2007.

[11] Kaiming He and Jian Sun. Image completion approaches using the statistics of similar patches. *TPAMI*, 2014.

[12] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Temporally coherent completion of dynamic video. *TOG*, 2016.

[13] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A Efros. Fighting fake news: Image splice detection via learned self-consistency. In *ECCV*, 2018.

[14] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ToG*, 2017.

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[16] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In *CVPR*, 2019.

[17] Sungho Lee, Seoung Wug Oh, DaeYeun Won, and Seon Joo Kim. Copy-and-paste networks for deep video inpainting. In *ICCV*, 2019.

[18] Haodong Li and Jiwu Huang. Localization of deep inpainting using high-pass fully convolutional network. In *ICCV*, 2019.

[19] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, 2018.

[20] Sifei Liu, Jinshan Pan, and Ming-Hsuan Yang. Learning recursive filters for low-level vision via a hybrid neural network. In *ECCV*, 2016.

[21] Yunqiang Liu and Vicent Caselles. Exemplar-based image inpainting using multiscale graph cuts. *TIP*, 2012.

[22] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *CVPR*, 2019.

[23] Babak Mahdian and Stanislav Saic. Using noise inconsistencies for blind image forensics. In *IMAVIS*, 2009.

[24] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do gans leave artificial fingerprints? In *MIPR*, 2019.

[25] Seoung Wug Oh, Sungho Lee, Joon-Young Lee, and Seon Joo Kim. Onion-peel networks for deep video completion. In *ICCV*, 2019.

[26] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[27] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[28] Mengye Ren and Richard S Zemel. End-to-end instance segmentation with recurrent attention. In *CVPR*, 2017.

[29] Ronald Salloum, Yuzhuo Ren, and C-C Jay Kuo. Image splicing localization using a multi-task fully convolutional network (mfcn). In *JVCI*, 2018.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[31] Chuan Wang, Haibin Huang, Xiaoguang Han, and Jue Wang. Video inpainting by jointly learning temporal structure and spatial details. In *AAAI*, 2019.

[32] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *CVPR*, 2020.

[33] Siyue Wang, Xiao Wang, Pu Zhao, Wujie Wen, David Kaeli, Peter Chin, and Xue Lin. Defensive dropout for hardening deep neural networks under adversarial attacks. In *ICCAD*, 2018.

[34] Siyue Wang, Xiao Wang, Pin-Yu Chen, Pu Zhao, and Xue Lin. Characteristic examples: High-robustness, low-transferability fingerprinting of neural networks. In *IJCAI*, 2021.

[35] Wei Wang, Jing Dong, and Tieniu Tan. Tampered region localization of digital color images based on jpeg compression noise. In *IWDW*, 2010.

[36] Qiong Wu, Shao-Jie Sun, Wei Zhu, Guo-Hui Li, and Dan Tu. Detection of digital doctoring in exemplar-based inpainted images. In *ICMLC*, 2008.

[37] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *CVPR*, 2019.

[38] Wei Xiong, Jiahui Yu, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo. Foreground-aware image inpainting. In *CVPR*, 2019.

[39] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *CVPR*, 2019.

[40] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018.

[41] Ning Yu, Larry S. Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *ICCV*, 2019.

[42] Ning Yu, Vladislav Skripniuk, Dingfan Chen, Larry Davis, and Mario Fritz. Responsible disclosure of generative models using scalable fingerprinting. *arXiv preprint arXiv:2012.08726*, 2020.

[43] Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *ICCV*, 2021.

[44] Markos Zampoglou, Symeon Papadopoulos, and Yiannis Kompatsiaris. Detecting image splicing in the wild (web). In *ICMEW*, 2015.

[45] Markos Zampoglou, Symeon Papadopoulos, and Yiannis Kompatsiaris. Large-scale evaluation of splicing localization algorithms for web images. In *MTAP*, 2017.

[46] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *ECCV*, 2020.

[47] Haotian Zhang, Long Mai, Ning Xu, Zhaowen Wang, John Collomosse, and Hailin Jin. An internal learning approach to video inpainting. In *ICCV*, 2019.

[48] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. In *CVPR*, 2018.

[49] Peng Zhou, Bor-Chun Chen, Xintong Han, Mahyar Najibi, Abhinav Shrivastava, Ser Nam Lim, and Larry S Davis. Generate, segment and refine: Towards generic manipulation segmentation. *AAAI*, 2020.

[50] Xinshan Zhu, Yongjun Qian, Xianfeng Zhao, Biao Sun, and Ya Sun. A deep learning approach to patch-based image inpainting forensics. *SPIC*, 2018.