Weakly-supervised deepfake localization in diffusion-generated images

Dragos-Constantin Tântaru
Bitdefender
dtantaru@bitdefender.com

Elisabeta Oneată
Bitdefender
eoneata@bitdefender.com

Dan Oneată
University Politehnica of Bucharest
dan.oneata@gmail.com

Abstract

The remarkable generative capabilities of denoising diffusion models have raised new concerns regarding the authenticity of the images we see every day on the Internet. However, the vast majority of existing deepfake detection models are tested against previous generative approaches (e.g. GAN) and usually provide only a “fake” or “real” label per image. We believe a more informative output would be to augment the per-image label with a localization map indicating which regions of the input have been manipulated. To this end, we frame this task as a weakly-supervised localization problem and identify three main categories of methods (based on either explanations, local scores or attention), which we compare on an equal footing by using the Xception network as the common backbone architecture. We provide a careful analysis of all the main factors that parameterize the design space: choice of method, type of supervision, dataset and generator used in the creation of manipulated images; our study is enabled by constructing datasets in which only one of the components is varied. Our results show that weakly-supervised localization is attainable, with the best performing detection method (based on local scores) being less sensitive to the looser supervision than to the mismatch in terms of dataset or generator.

1. Introduction

Image generation is improving by the day and it is arguably past the point where it is possible to perceptually distinguish between generated (fake) and real content. Generative adversarial models (GAN) [19], normalizing flows [46], denoising diffusion probabilistic models (DDPM) [54]—all provide excellent means for the creation of digital art or entertainment content. However, the advances in image generation come at the cost of also easing malicious use, e.g., by altering reality or spreading misinformation. To counter these harmful effects, deepfake detection methods are developed to discriminate between fake and real samples [43, 44, 56].

Among the classes of generative models, diffusion models are emerging as the dominant paradigm [14], showcasing impressive results on a wide array of tasks including text-controlled image generation [45, 48, 51, 61] or image-to-image translation [38, 48, 50, 61]. Prior work on deepfake detection has naturally mostly considered detecting content generated by GANs [5, 20, 41, 57, 60], but the computer vision community is now starting to consider DDPMs [9, 47]. Here we continue this direction, going one step further to address the task of weakly-supervised deepfake localization.

First, we extend prior approaches to localize the manipulated area and not only label the entire image as fake or real. The binary output of the typical deepfake detection methods provides only coarse and opaque information, especially in the frequent case of local manipulations and forgeries. In this scenario, we would be much better served by a richer representation that could pinpoint which part of the image is likely to have been generated. Another benefit of localization is that it allows the end-user to take more informed decisions. For example, changing the color of one’s eyes may just be an innocuous enhancement of the user’s appearance, but the alteration of the movement of the lips in a video may hint towards a malicious use. Instead of deciding upfront what is deemed to be fake or real, a localization method can defer this decision to the end user, who is more informed and can tailor the method to their use case.

Second, in contrast to prior work, which addresses localization in a fully-supervised setting [26, 33, 58, 64], we consider a weakly-supervised scenario, where we assume that we only have access to image-level labels and the models are not explicitly trained for localization. This setup is motivated by the fact that generative methods are usually first developed in the context of full-image synthesis, and only then extended to the more specific cases of local editing, such as inpainting or attribute manipulation. Moreover, ground truth manipulation masks might not always be available, especially for newly developed local manipulation methods. Training a deepfake localization method in a weakly-supervised fashion (based on a global label) would allow us to be one step ahead of the potentially harmful uses involving local changes.

Our work brings the following contributions:
1. We propose a **weakly-supervised** framework for deepfake localization in images that allows to systematically uncover the importance of various factors (**model, supervision type, dataset, generator**) in the context of weakly-supervised localization of face manipulations.

2. We generate a detailed dataset (more than 125k images) with locally- and fully-manipulated images that allows the disentanglement of different factors in deepfake manipulation localization. The images are obtained using either newly introduced state-of-the-art generative models or a novel inpainting approach that incorporates a pretrained LDM [48] model in a diffusion-based inpainting method [38].

3. We provide **extensive quantitative and qualitative results** to understand the fundamental factors underlying the performance of weakly-supervised localization models. Our analysis reveals the severity of out-of-domain degradation, provides insights into the model’s sensitivity to looser supervision or dataset mismatch, and quantifies the performance across multiple classes of generative models. Our code and dataset are available at https://github.com/bit-ml/dolos.

2. Related Work

**Deepfake detection of GAN content.** There is a vast and continuously-growing body of work dedicated to the detection of GAN-generated images, see [39, 43, 44, 56] for reviews. Prior research has revealed many particularities of GAN content [10, 20, 60], an important observation being the appearance of a fingerprint—an imperceptible pattern, which allows the identification of the GAN method and training dataset [41, 60]. Wang et al. [57] also observe that all CNN-generated images share common systematic artifacts, that can be easily picked up by a classifier, while in [20] the authors indicate that downsampling might destroy these high-frequency artifacts, which are the key to detection.

**Deepfake detection of DDPM content.** Preliminary works on detecting diffusion-generated images made use of high-level cues such as inconsistencies in lighting [17] or perspective distortion [18]. However, more common end-to-end detection networks were also tested on diffusion images [9, 47], focusing on the transferability across classes of generative models (from GAN to DDPM, and vice versa). The prevailing observation is that detectors trained on one type of data do not generalize well to the other, but finetuning helps.

**Local manipulations.** A common setup in deepfake creation is altering a person’s face by reenactment, replacement, editing or synthesis using techniques known as face swap, face transfer, facial attribute manipulations or inpainting [43]. These approaches result in local manipulations and are traditionally GAN-based. Increasingly larger and more complex datasets and challenges have emerged [15, 25, 28, 31, 34, 49] and, with these, a considerable effort has been made to expose those types of fakes [1, 3, 16, 23, 39, 63]. However, actually localizing manipulations has arguably received less attention than detecting whether an image is fake or not. Works that tackle localization rely on local noise fingerprint patterns [21, 33, 40, 64], attention mechanisms [12, 13, 42] or self-consistency checks [2, 27]. Very recent, concurrent works proposed a forensic framework for general manipulation localization [21] and a hierarchical fine-grained formulation for image forgery detection [22]. Similar to us they consider diffusion-generated data with local forgeries, but differently they assume full supervision.

3. Methodology

We first describe the methods used for deepfake detection and weakly-supervised localization (§3.1). Then we detail the generative techniques that we are interested in detecting (§3.2) and the datasets generated with these methods (§3.3).

3.1. Methods for detection and localization

The task of deepfake detection consists of predicting whether an image is either real or fake. This task is usually framed as a binary classification problem and it is addressed using standard classification networks. In this paper we are interested in evaluating the capabilities of such methods in a
weakly-supervised setting: if we assume only image-level labels, can these classifiers be successfully used for localization of partially manipulated images?

We identify and investigate three categories of architectures suitable for weakly-supervised localization. These methods are based on either explanations (GradCAM), local scores (Patches) or attention (Attention) (for visual depictions see Figure 1). The first category is a general technique that given a trained classification network it uses explainability techniques to highlight the most predictive regions for the “fake” label. The other categories implicitly construct the localization maps: Patches produces local patch scores that are then used for classification, while Attention predicts an activation map that is used to pool relevant classification features. To allow for a fair comparison we fix the backbone and, in particular, we select the Xception network [8], which has been shown to yield excellent results for deepfake detection of faces [49].

The proposed methods are inspired by and build upon state-of-the-art deepfake detection methods [5, 12, 49], but we further modify them as described below.

**GradCAM.** While GradCAM explanations were previously used in the deepfake detection literature [4, 53, 59, 63], they were mostly shown as qualitative results and rarely (if ever) evaluated quantitatively, in terms of how well they localize the input alterations. In this paper we aim to quantify their performance and contrast them with other weakly-supervised localization methods. Concretely, we endow the Xception network [8] network with localization capabilities by applying GradCAM [52] on the activations produced by block 11, the one before the last downsampling operation.

**Patches.** We use Patch–Forensics [5], which is a truncated image classification network: it takes the feature activations after a few layers and projects them to a patch-level score using 1 x 1 convolutions. At train time, the loss is computed for each patch, while at test time, it produces a detection score by averaging the per-patch softmax scores. The authors experiment with two backbones (Xception [8] and ResNet [24]) and vary the number of layers that are kept. We chose the Xception backbone truncated after the second block of layers, as this combination was shown to yield good performance [5]. One advantage of Patches is that its output naturally corresponds to a localization map. While visualizations of the activation maps were shown in the original work, the localization performance was not quantified.

**Attention.** We start from [12] which augments an Xception [8] backbone with a learned attention mask that is used to modulate the feature maps produced by the network. The network is trained in a multi-task setting, with a loss on the full-image fakeness score and another one on the localization mask. In the weakly-supervised scenario, when no groundtruth mask is provided, the second term ensures that the maximum value of the predicted mask agrees with the image-level label. We modify the original implementation in [12] to improve the performance and stabilize the training. First, we replace the L1 loss on the mask with the binary cross-entropy loss (CE). Second, we cross-validate the weight λ that balances the two losses. Our final loss is:

\[ L = CE(y, \hat{y}) + \lambda CE(y, \max \hat{m}), \]

where \( y \) is the true image label, \( \hat{y} \) is the fakeness score and \( \hat{m} \) is the estimated localization mask.

**Fully-supervised localization.** Along with the weakly-supervised setup we also consider the fully-supervised case to show an upper bound on the performance. Since not all considered detection methods are able to be trained in a fully-supervised setting out of the box, we modify them to accommodate this setup: for GradCAM we truncate after block 11 and add a fully convolutional layer as in [37]; for Attention we keep only the loss on the mask, otherwise the architecture remains the same; for Patches we maintain the same architecture, but instead of using the image label to supervise each feature prediction, we use the downsized mask as groundtruth.

### 3.2. Dataset generation methods

We use diffusion models to generate both full images and locally-inpainted ones.

**Diffusion denoising probabilistic models (DDPM) [52]** are a class of generative models trained to reverse a diffusion process. The forward diffusion process iteratively adds Gaussian noise to a sample until its distribution reaches a standard normal. The reverse denoising process gradually removes noise, producing novel samples when starting from a random image. The reverse process is implemented as a neural network (with parameters \( \theta \)) that predicts the mean \( \mu_\theta(x_t, t) \) and covariance \( \Sigma_\theta(x_t, t) \) of a Gaussian distribution:

\[ p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \]

where \( x_t \) are images that are sequentially generated, from \( t = T \) to \( t = 1 \).

**Repaint: Inpainting with diffusion.** The task of inpainting is to fill in the missing regions of an image \( x_0 \) such that the resulting composition looks natural; the missing regions are usually specified by a binary mask \( m \). For inpainting with diffusion we use the approach of Lugmayr et al. [38], whose method performs mask-guided decoding on any pretrained DDPM. More precisely, at generation time they first sample a new image \( \hat{x}_t \) from the previously-generated image, \( \hat{x}_{t+1} \), according to Equation (2), but then they replace the values of \( \hat{x}_t \) outside the given mask \( m \) with the values of the original image encoded after \( t \) steps \( x_t \):

\[ \hat{x}_t \leftarrow m \odot \hat{x}_t + (1 - m)x_t \]
We translate this idea to inpainting by running the low-dimensional latent space of a variational autoencoder (VAE). This procedure ensures that the values outside the mask are preserved from the original image $x_0$.

**Repaint–LDM: Inpainting with diffusion in the latent space.** Latent diffusion models (LDM) [48] have been shown to offer a scalable approach to generating high-fidelity images. Their main idea consists of performing diffusion in the (low-dimensional) latent space of a variational autoencoder (VAE). We translate this idea to inpainting by running the Repaint scheduler (Equation 3) in the latent space, $x \leftarrow \text{enc}(x)$, of the variational autoencoder and using an appropriately downsized mask, $\mathbf{m} \leftarrow \text{resize}(\mathbf{m})$. This procedure generates an (inpainted) latent code, $\hat{x}$, which is then inverted to the original pixel space using the decoder of the VAE, $\text{dec}(\hat{x})$. Notably, this method allows us to inpaint an image using any existing pretrained LDM model. To the best of our knowledge, this approach to inpainting is novel.

### 3.3. Datasets

To train and evaluate our models, we use real images and two types of fake images: fully-synthesized and locally-manipulated images. The datasets are summarized in Table 1 and examples are shown in Figure 2.

**Real data.** We use the CelebA-HQ and FFHQ face datasets as sources of real data. CelebA-HQ [29] consists of 30k images that were selected and processed from the CelebA dataset [36]; we keep the original splits for training, validation and testing. FFHQ [30] consists of 70k PNG images that have been crawled from Flickr and automatically aligned and cropped. Both datasets are popular choices for training generative models and, consequently, are suitable choices for training deepfake detection models. We select a subset of 9k train and 900 validation images from each of the two datasets to match the number of fake images that are generated.

**Fake data: Full-image synthesis.** We use the perception-prioritized (P2) diffusion method of Choi et al. [7] to sample fully-synthetic images. We chose this approach because (i) the authors provide pretrained models on the two real datasets mentioned above (CelebA-HQ and FFHQ), which enable a systematic experimentation, and (ii) the models are lightweight and hence the inference is reasonably fast. For both datasets we sample 10k images: 9k for training and 1k for validation. We do not evaluate on these fully-synthesized sets, hence no test set is provided. We refer to these datasets as P2/CelebA-HQ and P2/FFHQ, respectively.

**Fake data: Local manipulations.** We generate two locally-manipulated datasets using the Repaint method [38] to inpaint images from the CelebA-HQ and FFHQ datasets. We use the Repaint method on top of pretrained P2 models, namely its variants trained on CelebA-HQ and FFHQ, respectively. The inpainted regions correspond to various face attributes (skin, hair, eyes, mouth, nose, glasses). For CelebA-HQ, these annotations were manually labeled and are available in the CelebAMask-HQ [29] extension of the dataset, while for FFHQ these are obtained using a pretrained face segmentation method [32]. Given an image (corresponding to the identity of a person) we generate multiple inpaintings by randomly sampling masks corresponding to these face attributes and, for the smaller parts (eyes, mouth, nose), by also dilating them with a kernel of randomly-chosen size.

<table>
<thead>
<tr>
<th>Type</th>
<th>Family</th>
<th>Model</th>
<th>Dataset</th>
<th>Train</th>
<th>Val.</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>–</td>
<td>–</td>
<td>CelebA</td>
<td>9k</td>
<td>900</td>
<td>900</td>
</tr>
<tr>
<td>Real</td>
<td>–</td>
<td>–</td>
<td>FFHQ</td>
<td>9k</td>
<td>900</td>
<td>900</td>
</tr>
<tr>
<td>Fake full</td>
<td>Diffusion</td>
<td>P2</td>
<td>CelebA</td>
<td>9k</td>
<td>1k</td>
<td>–</td>
</tr>
<tr>
<td>Fake full</td>
<td>Diffusion</td>
<td>P2</td>
<td>FFHQ</td>
<td>9k</td>
<td>1k</td>
<td>–</td>
</tr>
<tr>
<td>Fake local</td>
<td>Diffusion</td>
<td>Repaint–P2</td>
<td>CelebA</td>
<td>30k</td>
<td>3k</td>
<td>8.5k</td>
</tr>
<tr>
<td>Fake local</td>
<td>Diffusion</td>
<td>Repaint–P2</td>
<td>FFHQ</td>
<td>30k</td>
<td>3k</td>
<td>–</td>
</tr>
<tr>
<td>Fake local</td>
<td>Latent diff</td>
<td>Repaint–LDM</td>
<td>CelebA</td>
<td>9k</td>
<td>900</td>
<td>900</td>
</tr>
<tr>
<td>Fake local</td>
<td>Fourier conv</td>
<td>LaMa</td>
<td>CelebA</td>
<td>9k</td>
<td>900</td>
<td>900</td>
</tr>
<tr>
<td>Fake local</td>
<td>GAN</td>
<td>Pluralistic</td>
<td>CelebA</td>
<td>9k</td>
<td>900</td>
<td>900</td>
</tr>
</tbody>
</table>

Table 1. Details of our proposed dataset, which contains locally- and fully-generated images from multiple types of generators. The dataset is designed to allow for a principled analysis of multiple factors: manipulation type, generator, source dataset. We provide: (i) fully-generated images on CelebA-HQ and FFHQ using P2 [7]; (ii) locally-inpainted images on FFHQ using Repaint–P2 and on CelebA-HQ using Repaint–P2, Repaint–LDM, Pluralistic [62], LaMa [55] (using the same masks).
but up to 15 pixels. We refer to the resulting datasets as Repaint–P2/CelebA-HQ and Repaint–P2/FFHQ; the former will represent our main test bed, while the latter is used only at training.

To be able to systematically study the importance of the generator we inpainted a subset of 9k images used in Repaint–P2/CelebA-HQ with three other methods: Repaint-LDM (ours), LaMa [55], Pluralistic [62]. Repaint-LDM adapts the Repaint method to operate in the latent space by using the LDM model [48]; LaMa is an inpainting method that uses an autonecoder with Fourier convolutions [6]; Pluralistic is a conditional variational autoencoder with adversarial loss. We have chosen these methods since they all provide pretrained models on the CelebA-HQ dataset. This allows us to inpaint the same images using the same masks, and isolate the differences attributed to the change of generator.

4. Experimental setup

Implementation details. Following the recommendation of Chai et al. [5], we ensure that real and fake images both follow exactly the same preprocessing steps prior to passing them through the detection methods. These steps include the input resolution and resize algorithm. Consequently, we process both CelebA-HQ and FFHQ images as they were processed for training the generator, that is, we resize them to 256 × 256 using bicubic interpolation.

Tasks and metrics. Localization is the main task that we tackle. We report intersection over union (IoU) and pixel-wise binary classification accuracy (PBCA). These metrics assume binary prediction and we use a fixed threshold of 0.5 for binarization. The detection methods generate masks of different sizes: 19 × 19 for GradCAM and Attention, 37 × 37 for Patches. For a fair evaluation we resize them to the size of the input image: 256 × 256.

We also report results on detection, the task of telling apart fake images from real images. We rank the images by their per-image fakeness score, which is output by each method as illustrated in Figure 1. The detection performance is then measured in terms of average precision (AP), which is a threshold-free metric.

5. Experiments

Our experiments evaluate the proposed methods with different levels of supervision, gradually changing the dataset and the generators in order to quantify their importance for localization. We investigate the performance using three main levels of supervision:

- **Setup A (label & full)** is a weakly-supervised setup in which we have access to fully-generated images as fakes and, consequently, only image-level labels. We use 9k fake images, fully synthesized by P2, and 9k real images from the corresponding dataset on which P2 was trained.

- **Setup B (label & partial)** is a weakly-supervised setup in which we have access to partially-manipulated images, but only with image-level labels (no localization information). This means that while an image may be labelled as “fake”, not all of its regions are fake. We use 9k locally-modified images by Repaint–P2 and 9k real images from the corresponding training dataset.

- **Setup C (mask & partial)** is a fully-supervised setting, in which we have access to ground-truth localization masks of partially-manipulated images. We uses 30k locally-modified images by Repaint–P2; for this setup, no real images are used.

To evaluate localization we use 8.5k locally-manipulated images produced by Repaint–P2/CelebA-HQ and to evaluate detection we use 900 real images from CelebA-HQ and 900 fakes from Repaint–P2/CelebA-HQ. Note that the evaluation is carried on the same data regardless of the setup. Table 1 from the supplementary material summarizes the data used in each of the three setups.

5.1. Evaluating localization abilities

We evaluate all three proposed approaches for localization in the three setups described above. To exclude other factors of variation we maintain the image generator and source
which the training and testing data come from different datasets. To this end, we design an experiment in multiple setups and metrics (bold values in Table 2). We observe strong performance of Patches generally outperforms the other two methods under most of the scenarios and metrics.

Table 2. Evaluation of the three selected localization techniques (GradCAM, GC, Patches PT, Attention AT) on the Repaint–P2/CelebA-HQ dataset using different levels of supervision: image-level label on full images (A), image-level label on locally manipulated images (B), and fully-supervised masks (C). We evaluate both localization (using IoU and PBCA) and detection (using AP). Patches systematically outperforms the other two methods under most of the scenarios and metrics.

<table>
<thead>
<tr>
<th>setup</th>
<th>sup.</th>
<th>generator</th>
<th>IoU (%)</th>
<th>PBCA (%)</th>
<th>AP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>GC</td>
<td>PT</td>
<td>AT</td>
<td>GC</td>
</tr>
<tr>
<td>A</td>
<td>label</td>
<td>full</td>
<td>16.8</td>
<td>64.9</td>
<td>9.7</td>
</tr>
<tr>
<td>B</td>
<td>label</td>
<td>partial</td>
<td>21.5</td>
<td>37.7</td>
<td>23.2</td>
</tr>
<tr>
<td>C</td>
<td>mask</td>
<td>partial</td>
<td>83.7</td>
<td>84.5</td>
<td>70.3</td>
</tr>
</tbody>
</table>

Among the selected methods, we see that Patches generally outperforms the other two approaches across multiple setups and metrics (bold values in Table 2). We see that localization performance is strong for all methods when training in the fully supervised scenario (setup C) and performance drops as we move to the two weakly supervised setups (setups A and B). Interestingly, GradCAM and Attention perform better in setup B than in setup A, while for Patches we observe the reverse trend. We believe that Patches perform worse in setup B because the loss is set at patch-level, and the patch labels are inherently noisy as we use partially-manipulated images at input.

In terms of detection (the ‘AP’ columns in Table 2), we observe strong performance of Patches in both weakly supervised setups, A and B. Interestingly, the detection performance is good for all models in setup B. In retrospect, this is expected since for the detection task in setup B the train data matches the test data.

Figure 3 shows examples of the localization maps produced by the detection methods in all three scenarios. We notice that Patches is able to partially recover the manipulated areas even in setups A and B. In setup B we observe that due to the noisy labels the model fires also on the background regions. GradCAM and Attention struggle more in the weakly-supervised scenarios and their outputs are qualitatively different: the former seems to produce weaker activations, which are spread through irrelevant areas of the image (especially in scenario A), while the latter produces less precise localizations.

5.2. Generalization across source datasets

Generalization is a key desirable property of deepfake detectors. Here, we assess how localization is affected by datasets shifts. To this end, we design an experiment in which the training and testing data come from different source datasets, while fixing the generator and the detection method. Training is either based on fake data derived from CelebA-HQ (P2/CelebA-HQ for scenario A and Repaint–P2/CelebA-HQ for scenarios B and C) or FFHQ (P2/FFHQ for scenario A and Repaint–P2/FFHQ for scenarios B and C), while the testing is carried on Repaint–P2/CelebA-HQ.

Quantitative results are shown in Table 3 for all scenarios under both localization and detection metrics. We observe a consistent drop in performance across all scenarios. This is more evident in scenario B where only image-level supervision is available for locally-manipulated images.

Table 3. Evaluation of Patches on the Repaint–P2/CelebA-HQ dataset using two training datasets: CelebA-HQ and FFHQ. When the source dataset does not match the target one, we observe a consistent drop in performance across all scenarios. This is more evident in scenario B where only image-level supervision is available for locally-manipulated images.

<table>
<thead>
<tr>
<th>sup.</th>
<th>generator</th>
<th>CelebA-HQ</th>
<th>FFHQ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IoU</td>
<td>PBCA</td>
</tr>
<tr>
<td>A</td>
<td>label</td>
<td>full</td>
<td>64.9</td>
</tr>
<tr>
<td>B</td>
<td>label</td>
<td>partial</td>
<td>37.7</td>
</tr>
<tr>
<td>C</td>
<td>mask</td>
<td>partial</td>
<td>84.5</td>
</tr>
</tbody>
</table>
We observe that localization works generally very well as long we test on data generated from the same model (main diagonal in the left plot). However, LDM is an exception: localization in LDM-manipulated images is more difficult since the inpainting is carried in the latent space and the decoding step “hides” the traces of the latent manipulation, akin to how image processing steps degrade detection performance [57].

When we evaluate on data coming from a different generator, the performance drops sharply (off-diagonal entries in the left plot). The transfer performance between LaMa and Pluralistic is still decent, presumably due to the particularities of the encoder–decoder approach. The diffusion model of Repaint is different from the two and makes the cross-generator transfer more challenging. Still, it appears that the transfer from diffusion to autoencoders and GANs (38.0% and 41.4%, respectively) is easier than the other way around (10.3% and 0.2%, respectively); a similar conclusion has been observed for detection [47].

Training on combinations of multiple datasets yields generally good performance on all the datasets involved at training (off-diagonal entries in the right plot). However, we do not observe a generalization benefit by using more types of generators at training (diagonal entries in the right plot vs. off-diagonal entries in the left plot). For example, training on all generators but LDM yields an IoU of 19.7%, which is only marginally above 19.5%, what is achieved by training only on Repaint. For the other three generators, the performance is even slightly worse on combinations than the single best generator.

5.4. Performance on unseen datasets

In this section, we consider generalization in its most challenging form, by varying both the source dataset and the generation algorithm. Consequently, we evaluate on a
different dataset, COCO Glide [21], which consists of 512 locally-edited images using a text-guided diffusion-based model. Additionally, we present results of five other existing localization methods [11, 21, 22, 35, 58], which were pre-trained on different datasets, and compare their performance on our own Repaint–P2/CelebA-HQ, as well as on COCO Glide. For a comparison to Patches, we also fine-tune the PSCC method [35] in setup C on the Repaint–P2/CelebA-HQ data. The results are shown in Table 4.

We observe that the generalization performance is modest on either of the two datasets: the best out-of-domain performance on Repaint–P2/CelebA-HQ is 23.1%, obtained by TruFor, while on COCO Glide is 33.3%, obtained by PSCC. Even methods that have shown to generalize (TruFor [21]) or that have been trained specifically on diffusion images (HiFi-Net [22]) have difficulties on out-of-domain datasets.

Patches shows competitive results (second best in terms of IoU on COCO Glide), even if it was trained solely on faces. Interestingly, this is not the case for PSCC. While PSCC obtains top performance in-domain, on Repaint–P2/CelebA-HQ, it struggles to generalize to COCO Glide. This behaviour suggests that overfitting is occurring, which is not surprising given that the model capacity of PSCC (3.6M parameters) is an order of magnitude larger than the one of Patches (200k parameters).

<table>
<thead>
<tr>
<th>Method</th>
<th>R-P2/CelebA</th>
<th>COCO Glide</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IoU</td>
<td>PBCA</td>
</tr>
<tr>
<td>MantraNet [58]</td>
<td>4.8</td>
<td>81.9</td>
</tr>
<tr>
<td>Noiseprint [11]</td>
<td>18.2</td>
<td>23.8</td>
</tr>
<tr>
<td>PSCC [35]</td>
<td>14.3</td>
<td>66.5</td>
</tr>
<tr>
<td>TruFor [21]</td>
<td>23.1</td>
<td>81.3</td>
</tr>
<tr>
<td>HiFi-Net [22]</td>
<td>0.0</td>
<td>81.0</td>
</tr>
</tbody>
</table>

Methods trained on Repaint–P2/CelebA-HQ in setup C

PSCC [35] 89.0 98.8 13.3 18.4
Patches 84.5 98.7 30.8 64.8

Table 4. Evaluation of pretrained localization models on our Repaint–P2/CelebA-HQ and the COCO Glide dataset [21]. The grayed out results (Patches and PSCC on Repaint–P2/CelebA-HQ) are not directly comparable to those of other methods, since both Patches and PSCC are trained on Repaint–P2/CelebA-HQ. Qualitative results are available in the supplementary material.

6. Conclusions

In this paper, we investigate weakly-supervised localization in the context of diffusion-generated images of faces. We propose a framework and a dataset that allows to systematically explore the importance of different factors in model performance, such as: choice of detection method, level of supervision, dataset and type of generator used. We design a series of experiments that progressively modify the training assumptions and showed that, to a certain extent, detection of local manipulations can be performed weakly supervised, even in the most restrictive scenarios.

We summarize our findings: 1. The patch-based method consistently outperforms the other two approaches (explanations or attention) across multiple settings and metrics. 2. The detection performance in one of the weakly-supervised settings (image label & partial manipulations) is strong across all detection methods, suggesting that partially-manipulated images can be used for training deepfake classifiers. 3. Among the three types of factors (supervision, dataset, generator method), supervision seems to have a lesser impact (at least for the best performing method, Patches), while the generator impacts the most. 4. Localization of manipulations for latent diffusion models is very challenging even in the most optimistic scenario.

We believe that these findings can fuel research into weakly-supervised localization of deepfake manipulations with possible extensions to general-content images and to other types of local manipulations, such as face-swap, local enhancements or facial pose transfer obtained with DDPMs.

Acknowledgements. This work was supported in part by European Union’s HORIZON-CL4-2021-HUMAN-01 research and innovation program under grant agreement No. 101070190 “AI4Trust”.
References


[64] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. In CVPR, pages 1053–1061, 2018. 1, 2