Scoop: Mitigation of Recapture Attacks on Provenance-Based Media Authentication

Yuxin (Myles) Liu
University of California, Irvine

Habiba Farrukh
University of California, Irvine

Ardalan Amiri Sani
University of California, Irvine

Sharad Agarwal
Microsoft

Gene Tsudik
University of California, Irvine

Abstract

Continuous advances in photo and video manipulation yield increasingly sophisticated deepfakes that greatly endanger societal perception of reality. Deepfake detection is an intuitive and natural research direction, which is unfortunately shaping up to be a never-ending arms race. An alternative promising direction is provenance assertion, which blends hardware-based secure camera design with the cryptographic means of authenticating the source of visual content and any post-processing (e.g., filters) applied to it.

This work starts by highlighting a very effective attack type, called a *recapture attack*, against all provenance-based techniques. In such an attack, the adversary displays fake content on some form of a screen (e.g., TV, projector, or computer screen) or surface (e.g., cardboard, canvas, or paper) and uses a provenance-asserting secure camera device to capture photos and videos of the displayed content.

We then introduce Scoop,[1] a systematic solution for mitigating recapture attacks. Scoop leverages state-of-the-art depth sensing technologies as well as learning-based depth estimation to detect misleading recaptures, i.e., a recaptured photo or video where the presence of a display medium is not visually identifiable.

We implement Scoop on both iOS and Android platforms (Apple iPhone 14 Pro and Samsung Galaxy S20 Plus), using their built-in depth sensors. To evaluate the effectiveness of Scoop, we construct a first-of-its-kind dataset consisting of 78 recapture attack scenarios. Our results show that Scoop achieves as high as $\approx 95\%$ accuracy on the iPhone and 74% accuracy on the Samsung phone.

1 Introduction

Today, digital media is constantly produced and consumed in enormous volumes. It is an undisputed fact that modern society is dependent on and, to some degree, even addicted, to it. Due to the global popularity and affordability of digital cameras and camera-equipped smartphones (and, increasingly, eye-wear and AR/VR headsets), almost everyone can create visual content (photos and videos) anywhere anytime. Furthermore, ubiquitous Internet connectivity allows visual content to be easily shared online. It is estimated that over 3.2 billion images and 720,000 hours of videos are shared daily on various social media platforms [71].

Figure 1: *Demonstration of effectiveness of recapture attacks. One of these photos is real (i.e., captured by a camera in a real environment) and one is a recapture attack (i.e., captured by a camera pointed at a screen showing a modified photo). Even though we have used a mid-range TV, detecting the attack is very challenging (especially if the user is not suspecting an attack). Appendix A mentions which one of these photos is real.*

On the other hand, besides recreational purposes, visual content can benefit the entire society, e.g., via news reporting (both mainstream and citizen journalism), legal proceedings, law enforcement actions, and so on. Over one quarter of American adults get their news from YouTube, and over half – from social media platforms [39,70]. Also, about 65% of American **X** users cite reading news as their main reason for using the platform [38].

Unfortunately, lots of visual content found on the Internet is not real, ranging from manipulated to fully synthetic. Continuous advances in image and video editing techniques and deepfake creation [76] make it very hard for consumers to

[1] Scoop: **S**ecure **C**ontent **O**rigin from **O**ptical **P**roperties

determine visual content's veracity. To make things worse, recent development of generative artificial intelligence (AI) technologies [23, 30] enable deepfakes to produce arbitrary lifelike photos and videos with just a few lines of text prompt.

The above, coupled with the popularity of social media sharing platforms and human gullibility, results in rapidly growing volume and efficacy of mostly nefarious misinformation [1, 9, 67, 68, 73]. One notable example is a manipulated video of the Ukranian president Zelinsky, publicly addressing the war between Russia and Ukraine [40]. Misinformation impacts organizations and institutions (industry, non-profit, and government) as well as individuals.

The unprecedented rise in misinformation attributed to deepfakes triggers an urgent need for authenticity and integrity of visual content. To this end, both industry and academia developed various provenance-based techniques [5, 7, 45, 57, 58, 60]. The key idea behind this line of work is the use of cryptographic metadata attached to visual content, coupled with a secure camera design. The end-result is the ability for anyone to authenticate the source camera, the content it captured, and any post-processing filters applied to that content. Based on secure components, such as a Trusted Execution Environment (TEE) [3, 57] and specialized hardware, such as a secure camera module (sensor) [58], provenance-based techniques aim to provide strong security for generation and subsequent (benign) modification of visual content.

Many prominent industry players from diverse sectors (e.g., Canon, Adobe, BBC, and Microsoft) already committed to – or even commercialized – provenance-based media authentication techniques [5]. For example, Truepic teamed up with Qualcomn to provide provenance-based media authentication for smartphones with Qualcomm Snapdragon SoCs [26]. Another example is Adobe's recent integration of content authenticity into its tools [16].

This paper sheds light on an important vulnerability in provenance-based techniques: vulnerability to *recapture attacks*. In such an attack, the adversary prepares some fake content, which it then either: (1) displays on a screen, e.g., TV, monitor, or projector, or (2) prints or replicates on some physical surface, e.g., paper, cardboard, or canvas. Finally, the adversary captures this fake content with a camera that uses a provenance-based technique. The fake content might comprise the entire frame or part(s) thereof.

The adversary's goal in a recapture attack is to present all captured content as real, including the fake parts. Since a provenance-based camera cannot distinguish real content from displayed, painted, or printed one, the entirety of captured content gains credibility as having been generated by a secure camera. Figure 1 demonstrates the effectiveness of this attack. It shows two photos, both of which are captured by the same smartphone. One of these photos is captured in the real physical environment depicted in the photo. The other is a fake, manipulated version of the original photo displayed on a TV and recaptured by the same smartphone. Detecting the malicious photo is very challenging, if not impossible, especially if the viewer is unaware of the attack. To demonstrate this vulnerability, we conduct a user study. In §2.3, we present the results of this user study that show that people are not able to to distinguish between an original photo and a photo recaptured from a digital screen.

To mitigate recapture attacks in general, we introduce Scoop[2]. It uses depth information captured at the same time as the visual content with a time-of-flight (ToF) sensor present on the camera to detect flat display mediums that could be used to mount recapture attacks (e.g., TVs, printed cardboard cutouts, and projector screens). Scoop tries to detect *misleading* recaptures, i.e., a recaptured photo or video where the presence of a display medium is not visually identifiable. We note that not all misleading recaptures are malicious, hence we intend Scoop as an assistant tool, helping the user detect recapture attacks.

Although the use of depth may seem intuitive and/or trivial, it poses an important challenge: the majority of flat surfaces are not display mediums that can be used in recapture attacks, e.g., walls and floors. To address this challenge, Scoop computes a learning-based depth map of the scene, which provides us with the perceived depth in the same photo. When the perceived depth map fails to match the real depth map generated by the ToF sensor, Scoop marks that region of the photo as *misleading*, prompting the viewer to pay more attention to it.

To evaluate the efficacy and practicality of Scoop, we assemble a first-of-its-kind dataset, which contains 122 unique data points, including 78 recapture scenarios and 44 benign, ordinary daily scenes.[3]

To demonstrate Scoop's viability and practicality, we built a prototype composed of two parties: producer (camera) and consumer (viewer). The former runs on both Apple iPhone 14 Pro (w/ dToF sensor) and Samsung Galaxy S20 Plus (w/ iToF sensor), while the latter runs on a Linux desktop. Evaluation results show that Scoop achieves up to 94.81% accuracy of detecting with as low as 0.02% false positives on an iPhone and 74.03% accuracy and 17.78% false positives, respectively, on a Samsung phone.

This paper makes the following contributions:

- Discussion and analysis of recapture attacks against provenance-based techniques.
- Design of Scoop, a technique that blends depth and visual information and can effectively detect misleading recaptures using flat display medium.
- A first-of-its-kind dataset containing 122 unique data points (both photos and videos) including 78 recapture scenarios, serving as a cornerstone for evaluating systems such as Scoop.

[2]We will open source Scoop upon publication.

[3]We will release this dataset so that it can be used for evaluating future methods that tackle the same problem.

Figure 2: *An example of a recapture attack on a provenance-based secure camera.*

- A fully functional Scoop prototype on commodity devices and a thorough evaluation thereof, including effectiveness, performance and storage overhead, and energy consumption.

2 Recapture Attack

Workflow and example. Figure 2 illustrates a recapture attack. The photo on the left is real and captured by a smartphone. The one in the middle is a recapture attack. To conduct this attack, the adversary digitally modifies the original photo to remove the wallet, displays the modified photo on a TV screen, and recaptures it using a provenance-based secure camera. The result is a fake photo that looks very real and, even worse, is endorsed by a provenance-asserting camera, creating a false and dangerous sense of authenticity for the viewer.

Admittedly, such attacks raise the bar for the adversary in terms of the amount of effort required. Purely digital fakes can be easily generated in large quantities by software tools with minimal human involvement. In contrast, recapture attacks require the adversary to physically stage the scene. However, the cost is not prohibitive since recapture attacks can be orchestrated with minimal equipment (e.g., a TV) and relative ease (e.g., mounting a phone in front of a TV).

History. Recapture attacks are not a new concept [37, 42, 50, 53, 72, 74]. In the past, these attacks have been used to deceive face authentication systems (aka face spoofing), since most early facial recognition systems lacked liveness detection and/or depth-based defense means [59]. Printed photos were a common way of mounting such attacks [59]. However, the display technology evolved to the point where a screen could display images with the same quality as that of printing. Plus, digital displays support high-definition video playback. Consequently, screens became the dominant tool used in recapture attacks [35]. Much effort has been made to mitigate these attacks, and some countermeasures are already deployed commercially [11, 27, 43, 52, 56].

Why relevant now? We found that many current countermeasures focus on one specific problem (i.e., face authentication), rather than providing a systematic approach to mitigating recapture attacks. One important reason is that recapture attacks have only targeted specific systems, such as spoofing face authentication systems. Indeed, in the past, there was no need for recapture attacks to deceive people with some fabricated information, since no provenance-based techniques were deployed and the adversary could simply present fake visual content.

2.1 Categorizing Recapture Attacks

The envisioned recapture attack is conducted by displaying fake content on some display medium and capturing that with a provenance-asserting camera.

One component needed to mount the attack is the *display medium* – the physical medium used to display fake content. We identify three categories for the display medium:

1. Screen-based (e.g., LCD/OLED TVs)
2. Projection-based
3. Print-based (e.g., Inkjet/laser/3D printing)

Moreover, the display medium could be *flat*, *curved*, or custom-shaped (e.g., a 3D-printed model).

(a) **Full-Recapture** (b) **Partial-Recapture**

Figure 3: *Illustration of full-recapture (a) and partial-recapture (b) attacks. Face in the image is blurred for anonymity.*

We also categorize recapture attacks into *full-recapture* and *partial-recapture* attacks, which are demonstrated in Figure 3. In the former, all contents (i.e., pixels) captured with the camera are recaptured, e.g., the camera points directly at a TV screen and captures part of the screen, while the TV is displaying fake content. In the latter, captured contents (i.e., pixels) contain both genuine and recaptured information, e.g., the camera points at a TV, while a real person is standing in front of the TV screen. In this example, the screen could display a place of interest, which would create a false sense of the person being physically present at the displayed location.

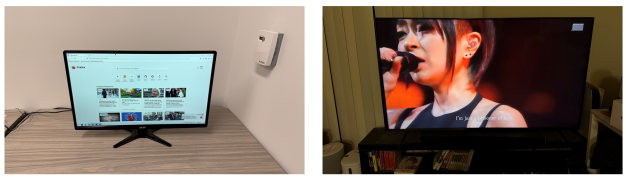


Figure 4: *These two example photos, even if display mediums (i.e., a monitor and a TV) are presented in them, do not constitute an attack since the presence of the medium can be recognized by the viewer.*

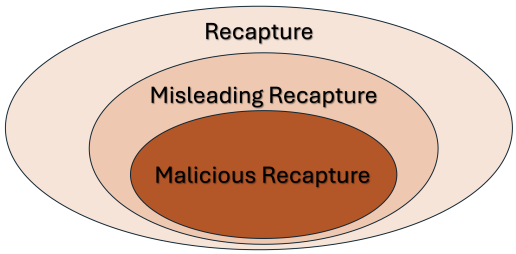


Figure 5: *Terminology of recaptured visual content. Note that Recapture is the largest set, with Misleading Recapture as a subset of it, followed by Malicious Recapture as a subset of Misleading Recapture.*

2.2 Terminology

We categorized recaptured attacks based on the type of display medium. However, the presence of one of the aforementioned mediums in a photo/video does not constitute an attack if the viewer can clearly recognize it. Moreover, whether a recapture is an attack or not depends on the content, i.e., whether it has been maliciously created. In this subsection, we more accurately define the terms we use to refer to recaptures.

We define a photo/video to be a *recapture* if it is taken off a display medium (fully or partially). However, sometimes, the presence of the display medium might be visually identifiable. This happens when the frames/boundary/edges of the medium are clearly visible. Figure 4 shows two such examples.

When the display medium is not visually identifiable, we refer to the recapture as *misleading* since the viewer might not realize that the content is recaptured. Note that a misleading recapture is not necessarily malicious. For a misleading recapture to count as an attack, the content shown on the display medium must be maliciously crafted in order to fool the viewers in some way. For example, consider a user taking a photo of a large TV showing some important event (a *full-recapture*). Viewers of this photo might be misled into thinking that the photo was taken during the event. However, this photo is not maliciously crafted and does not intend to be an attack. Figure 5 illustrates the aforementioned terminology.

As mentioned earlier, Scoop's goal is to identify misleading recaptures and highlight them to the viewer, enabling the viewer to further analyze them in order to detect attacks.

2.3 User Study

As mentioned, Scoop's goal is to identify misleading recaptures. A key question is whether users themselves are able to detect misleading recaptures or not. In other words: are today's display mediums and cameras capable of creating a misleading recaptured photo/video with adequate quality that goes undetected by a human viewer?

We conduct a user study to answer this question. In the user study, we only focus on using large, high-quality TVs (see § 8 for the TV models) as the display medium. While we believe high-quality projectors and printed cardboard cutouts can also go undetected by viewers, we do not own such display mediums in order to test that hypothesis. In addition, we focus on *full-recaptures*. This is because we deem *full-capture* attacks as the most common and effective recapture attack at the moment. We do however believe that *partial-recaptures* can also be effectively used to mount attacks, but we do not evaluate that in this user study.

In the user study, we showed 16 photos (8 original and 8 recaptured) to each user in a random order. Photos are selected to cover as much diversity as possible, i.e., lighting conditions, distance to objects, and complexity of the scenes, where all of them can be found in Appendix B. We recruited a total of 43 adult participants from the authors' institution. They have an average age of 24 years old, where 28 of them are in graduate degrees, and 15 of them are in undergraduate degrees. Among all participants, 25 of them are male and 18 of them are female.

Each session of the user study was conducted in person with one participant. All participants used the same device: a laptop featuring a 14-inch 2K resolution screen. To minimize potential psychological stress, the user study was unsupervised, where each participant took the survey with the laptop in an empty room using the QuestionPro [4] platform. The survey first provided each participant with the definitions of an original photo and a recaptured photo. Then for each photo in the survey, it gave them two choices to select from: original and recaptured. At the end of the survey, we asked participants about their experience completing it and confidence in their responses.

The results prove the fact that original and recaptured photos are perceptually indistinguishable. To demonstrate it, we conducted a one-sample t-test comparing our mean accuracy against chance performance. The results show that participants' (correct classification) accuracy 50.15% ($SD = 13.89\%$) is close to pure chance at 50% ($t(42) = 0.071$, $p = 0.944$), where the 95% confidence interval is between 45.88% and 54.42%. The negligible effect size ($d = 0.011$) indicates that any deviation from chance performance is practically meaningless. We also note that 44% of our participants classified exactly 8 out of 16 photos correctly. Additionally, most participants of our survey also stated that they did not have any confidence when they were making their decisions.

Finally, we note that in the user study, the participants were made aware of the presence of misleading recaptures and yet could not identify them. In realistic attack scenarios, users will be unsuspecting and hence are very unlikely to identify a misleading recapture.

3 Threat Model

There are two entities in Scoop interacting with photos or videos: producer (camera owner/operator) and consumer (content viewer). We assume the adversary to be the former, who might try to capture malicious photos or videos in order to deceive the latter who is the victim here. The adversary uses a provenance-asserting camera to take a photo/video of a display medium showing fake content. The adversary can show arbitrary fake content on the display. They cannot, however, modify the final photo/video captured by the provenance-asserting camera. Adversary's attempts to tamper with the captured content will be detected in the provenance verification process. (Please see §10.5 for a discussion on some limited form of provenance-based post-processing that could be allowed on the final photo/video.) The adversary cannot mount any software or hardware attacks on the camera device either, which is deemed trusted (e.g., secured with a trusted camera sensor [58]). In other words, we assume that the captured content's provenance information can be reliably and securely generated. This assumption is based on significant advancements in minimizing the trusted computing base (TCB) for provenance-based media authentication systems by both academia [57, 58] and industry [5, 7, 26] over recent years.

In this paper, we focus on recapture attacks using flat display mediums (see §2.1). Attacks using curved or custom-shaped display mediums are out of our scope. We note that our solution is likely able to handle curved display mediums. However, we do not have any samples of such attacks in our dataset and hence cannot evaluate our approach. (But we note that even the flat display mediums we use in our dataset are not completely flat and they can be slightly curved, textured, or bumpy.) On the other hand, our solution, as it stands, fails against custom-shaped medium (e.g., a 3D-printed model) since the depth map of the medium matches the learning-based perceived depth map. (Please see §10.1 for a more detailed discussion on 3D display-based attacks.)

In addition, we consider any objects outside of the supported range of the depth sensors as out of scope as well. Existing ToF sensors on smartphones have limited range – about 8 meters or 26 feet in our experiments. Those on autonomous vehicles have much higher range. While not a fundamental limitation of Scoop, we believe the ≈ 8 meter range is sufficient to detect most indoor misleading recaptures, as the cost of very large screens to mount attacks beyond 8 meters is very high. We hope that Scoop inspires smartphone vendors to include more powerful ToF sensors. Some professional-grade cameras have ToF sensors with more range, such as the DJI Zenmuse L2 with a range of 450 meters or 1476 feet [29].

Finally, we note that Scoop cannot distinguish between a flat surface such as a wall, and a display showing an image of a wall. In such situations, both the ground truth depth map and the learning-based depth map would match. We do note that meaningful attacks could exist in such scenarios, such as a recapture of a digitally modified signed contract. Scoop is not capable of detecting such recaptures.

4 Overview

Scoop tries to detect misleading recaptures. These recaptures can then be brought to the attention of the viewer, who can determine whether they are malicious or not.

Our first key idea to detect recaptures is to use a Time-of-Flight (ToF) depth sensor, available in some modern smartphones (almost all newer iPhone models and some Android devices) and cameras, to capture depth information of the scene. ToF depth sensor is a kind of camera sensor that measures the distance between the sensor and objects in front of it, thus providing a precise depth map [25]. Since we focus on recaptures that use flat surfaces, the idea is to use the ToF sensor to detect misleading recaptures.

However, mere use of depth information to mitigate recapture attacks faces an important problem: the existence of many non-display flat surfaces (e.g., walls). Scoop's goal is to detect display mediums, but not these other flat surfaces. To do so, Scoop computes a learning-based depth map of the scene. By comparing this map with the one generated by the ToF sensor, Scoop can detect content that has visual depth information and is shown on a display medium.

Insights behind our approach. We begin by looking at how and why recapture attacks work. They work by trying to make the viewer believe that the content shown in the photo/video is captured at its original scene rather than on a display medium, such as a poster or a TV screen. These attacks mainly abuse the fact that the viewer can only observe the captured content at the viewpoint decided by the photographer, i.e., in 2D. Moreover, since both printing and digital displaying technologies quickly advance, it becomes hard for a human eye to tell if they are looking at a real scene or a photo/video being shown on a medium at certain angles. Therefore, certain recaptures can mislead viewers into believing that a fake scene is real, i.e., has depth. As a result, detecting misleading recaptures requires us to understand human perception of the depth of an image. The main idea here is to achieve that using AI.

As is well known, AI has been dramatically growing in popularity in recent years. Various learning-based models have advanced to the point where they perceive information akin to humans, including monocular depth estimation, which is a kind of model that estimates depth using only a single photo containing RGB information.

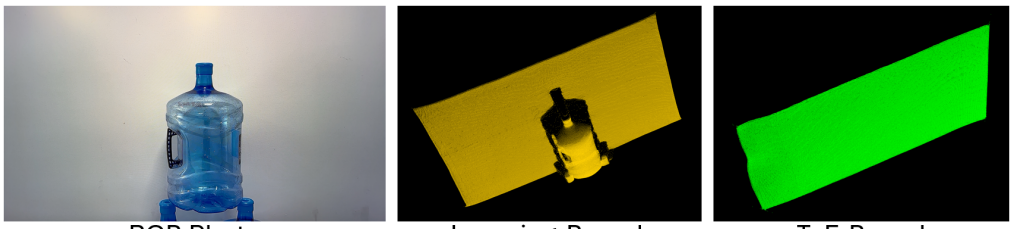


Figure 6: *Illustration of a learning-based depth estimation model got tricked by a photo displayed on a TV, while a ToF sensor captured depth map correctly represents the depth of the real scene.*

After some initial experiments, we found that, similar to humans, such models can be easily tricked by recapture attacks. For instance, we conducted an experiment where an RGB photo of a real scene was displayed on a TV screen and then was captured with: (1) TV screen frame being visible (i.e., a non-misleading recapture) and (2) without it being visible (i.e., a misleading recapture). In (1), depth estimation models did not recognize any depth in the photo, just a flat surface. In contrast, in (2), all models [32–34, 46, 49, 62, 63, 77–79] estimated depth information based on the content displayed on the TV screen.

While monocular depth estimation models fail to recognize a recapture attack in (2), the ToF depth sensor can recognize that there is no depth in this scene, since it measures the physical distance between the camera and each point in the scene. Figure 6 illustrates the depth maps generated using monocular depth estimation model and the ToF sensor.

Based on the above observations, our approach involves comparing the depth information generated by the learning-based method and the ToF sensor. If there is a significant discrepancy between the two, it is likely that we have identified a misleading recapture.

Alternative approaches. One might wonder whether providing verifiable provenance for time and location of content might be adequate to detect misleading recaptures. We note that these two types of metadata can help identify some misleading recaptures, but they are not always useful, i.e., when the content itself does not suggest a clear time or location. Moreover, providing authentic time and location metadata in the camera is not straightforward. For time, the camera needs to synchronize its clock to an external reference server. For location, it needs to use an external signal, e.g., GPS. However, GPS is unavailable for indoor scenarios and vulnerable to spoofing through low-cost tools [64, 75, 80].

5 Depth Estimation Preliminaries

We now summarize two main building blocks used in Scoop: ToF depth sensor and learning-based monocular depth estimation.

5.1 ToF Depth Sensor

ToF sensor, also known as ToF camera, is a special camera sensor used for measuring distance between the camera lens and the measured object. Similar to traditional camera sensor, it usually has multiple pixels, where each pixel can produce a distance value. Combining all pixels together, a depth map can be produced to recreate the 3D scene that the ToF sensor is seeing. ToF sensor can achieve high precision to the range of centimeters or even millimeters on some high-end models [31]. Also, because a ToF sensor works by detecting light it emits, it works well under low-light conditions.

There are mainly two types of ToF sensors: direct time-of-flight (dToF) sensors and indirect time-of-flight (iToF) sensors. A dToF sensor works by directly counting the time difference between the time that the sensor emits the light and the time that the sensor receives that light. Apple devices, such as iPhone 14-Pro, use a dToF (i.e., LiDAR) sensor [12]. An iToF sensor works by comparing the phase of emitted light and received light, where the phase difference is used to derive the distance value. Some Android devices, such as Samsung Galaxy S20 Plus, use an iToF sensor [6]. Due to the technical principle of a iToF sensor, it is more vulnerable to environmental lights and is also more likely to get confused when its emitted light bounce around before coming back [8, 28, 44].

5.2 Learning-based Monocular Depth Estimation

Monocular depth estimation is the ability to determine distance to each pixel in a photo with only a single RGB image. With the rise of AR/VR and autonomous driving, it has become a hot research topic in computer vision community in recent years. Its task is usually carried out by a machine learning model, trained on a large number of datasets with various scenarios and their corresponding depth maps.

The task has also been classified into two categories: relative and absolute depth estimation. The former aims to give the relative depth relationship among multiple objects in a photo, while the latter provides a precise estimation of the distance value for each pixel in the photo. Recent advances in absolute depth estimation can achieve centimeter-level

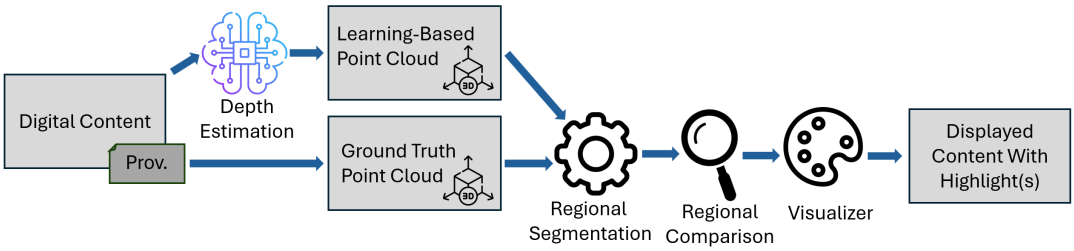


Figure 7: *Scoop viewer's workflow.*

accuracy [34]. However, just like human beings, the learning-based approach sometimes make mistakes, especially on flat surfaces showing content.

6 Design

We design Scoop to be integrated with both existing camera pipelines and provenance-based media authentication systems [5, 7, 57, 58]. In Scoop, in addition to the ordinary RGB information, the camera also captures depth information with its ToF sensor and embeds it as part of the final captured content's provenance information. The embedded depth information is essentially another photo containing the same scene of the RGB photo, but it keeps an absolute metric depth value for each pixel instead of a RGB value.

Whenever the captured content (a photo/video) reaches the viewer side, it can be opened with any content viewer, but the embedded depth information is only utilized when the content is opened with a Scoop-compatible viewer. During content playback, Scoop reads the depth information and generates a ground truth point cloud. At the same time, it also uses the corresponding RGB information to produce a depth map with a learning-based monocular depth estimation model. After that, the depth map is used to generate another point cloud. Scoop then compares these two point clouds against each other with Scoop comparison techniques. If any mismatch is found between the two point clouds, the content viewer highlights the mismatch to warn the users.

Note that Scoop only warns users of misleading (i.e., potentially malicious) recaptures by highlighting the part(s) of the frame flagged by its techniques. We made this design choice as we do not view Scoop as a determination tool of telling users if the digital content should be trusted or not, but as an analysis tool to assist users to make their own decisions. To better understand if this approach would work out or not, a future user study needs to be conducted. Another solution we have in our mind is to have a third party service running Scoop and make a decision, which helps offloading users' need to analyze Scoop's highlights. We also note that advanced reasoning AI could be a great fit to fulfill the need to analyze if Scoop's highlights may pose serious misinformation.

Figure 7 shows the entire workflow of Scoop viewer. We discuss more technical details of Scoop in §7.

Algorithm 1 Scoop's depth map comparison algorithm

Inputs: $\textbf{depth}_{learning}$, $\textbf{depth}_{truth}$
Output: **violated_pRegions**

Create $\textbf{pCloud}_{learning}$ with $\textbf{depth}_{learning}$
Create $\textbf{pCloud}_{truth}$ with $\textbf{depth}_{truth}$
Extract $\textbf{pRegions}_{truth}$ from $\textbf{pCloud}_{truth}$ using regional segmentation
Initialize **violated_pRegions**
for Every $\textbf{pRegion}_{truth}$ in $\textbf{pRegions}_{truth}$ **do**
    if Technique_1 (direct comparison) with $\textbf{pRegion}_{truth}$ and $\textbf{pCloud}_{learning}$ returns true **then**
        Add $\textbf{pRegion}_{truth}$ to **violated_pRegions**
    else
        Continue
    end if
    if Technique_2 (deviation correction) with $\textbf{pRegion}_{truth}$ and $\textbf{pCloud}_{learning}$ returns true **then**
        Add $\textbf{pRegion}_{truth}$ to **violated_pRegions**
    else
        Continue
    end if
    if Technique_3 (transformation) with $\textbf{pRegion}_{truth}$ and $\textbf{pCloud}_{learning}$ returns true **then**
        Add $\textbf{pRegion}_{truth}$ to **violated_pRegions**
    else
        Continue
    end if
end for
Return **violated_pRegions**

7 Depth Maps Comparison

As stated in §6, when the digital content is opened with an Scoop-compatible viewer, Scoop generates two point clouds (for each photo or video frame): a ground truth point cloud based on the ToF sensor captured depth map and a learning-based point cloud based on the estimation model perceived depth map using RGB information. Scoop then compares these two point clouds by first performing regional segmentation on the ground truth point cloud, followed by correlating each of the ground truth regions with its pixel-level corresponding part in the learning-based point cloud using multiple techniques: direct comparison, deviation correction, and transformation. Algorithm 1 shows the algorithm used by Scoop, where details are explained in the below subsections.

7.1 Regional Segmentation

The technical goal of Scoop is to find mismatches between the two point clouds. Unfortunately, this is not as easy as calculating the difference between two point clouds. First of all, although ToF sensors can produce depth maps with

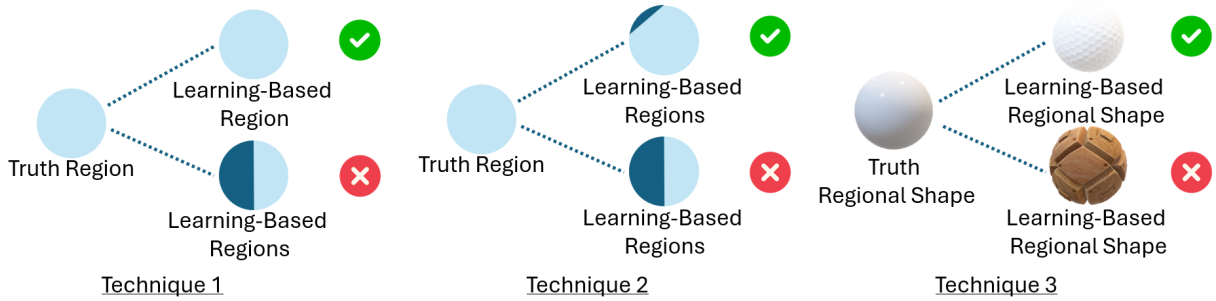


Figure 8: *Scoop viewer's techniques. In each technique, we try to match the region extracted from the ground truth point cloud with its corresponding (pixel-level correspondence) region in the learning-based point cloud. Only if it cannot be matched, we move to the next technique, until we run out of techniques, where we would flag the region as suspicious.*

high precision, their resolutions are usually relatively low, especially when compared with RGB camera sensors; additionally, noises are common in the captured depth map. On the other hand, learning-based point clouds can achieve high resolutions, but their precision is usually relatively low, especially when compared with ToF sensors' captured depth maps. Even if we match their resolutions by scaling, their depth values are still drastically different from each other, and the difference changes across the same photo. In our observation, the learning-based point cloud usually represents the relative depth of a single object well in one photo, but when combined with other objects and backgrounds, its precision drops significantly.

To overcome these challenges, Scoop does not blindly compare absolute depth values across all pixels, but instead it compares by regions. After initial noise reduction is applied to both point clouds, Scoop uses a *region growing algorithm* to perform cluster segmentation on the ground truth point cloud. We choose a region growing algorithm due to the nature of surfaces, where it is hard to find any perfectly flat ones. Even if this paper focuses on display mediums that are flat, our algorithm still needs to compute on depth information of non-flat surfaces and objects in the photo. We think the region growing algorithm is well suited for our use cases and can correctly separate all different surfaces existing in one scene. After the first segmentation, Scoop applies the same algorithm to the learning-based point cloud's corresponding regions (i.e., regions created with the previous segmentation on the ground truth point cloud) as well, which is detailed in §7.2

We note that parameters for performing cluster segmentation varies across depth maps for different capture/generation methods and scenes. For the former, difference in parameters is almost always fixed, where for the latter, parameters might need to be specially tuned for different objects. For example, human bodies tend to be continuous and could be recognized as a single surface with our default parameters, where stricter thresholds need to be set for correctly segmenting human bodies. In this case, learning-based image understanding models [21] could be used to first recognize different types of objects in the photo and then pick the best parameters for conducting cluster segmentation.

7.2 Regional Comparison

After performing cluster segmentation, we end up with many regions in the ground truth point cloud. For each of these regions, we compare it with its counter part in the learning-based point cloud, where the counter part is extracted by matching X and Y axes of all individual pixels in the region. We use multiple techniques to perform region comparison, where our goal here is to understand the similarity between them, and the reason we apply multiple of them is that there is no single technique that can cover all kinds of scenes, and there are also always imperfections in cluster segmentation, which could render some of them unusable. There are three techniques in Scoop (as illustrated in Figure 8): (1) direct comparison, (2) deviation correction and (3) transformation.

In Technique (1), we perform cluster segmentation again on the learning-based point cloud's part to check if we are only getting one region, which tells us that the learning-based point cloud agrees with the ground truth point cloud in this region. In Technique (2) (which is used when Technique (1) cannot be satisfied), we then try to relax the standard for checking. For example, we further examine if the biggest sub-region covers more than a certain threshold (e.g., 90%) percentage of the entire region (that is being checked). In Technique (3) (which is used when both Techniques (1) and (2) cannot be fulfilled), we use the iterative closest point (ICP) [19] and clustered viewpoint feature histogram (CVFH) [47] algorithms to perform the last attempt for getting the two regions matched with each other. The ICP algorithm tries to register one point cloud in another by transforming depth points, where once it succeeds, we check its convergence score to see if it falls below our threshold. On the other hand, the CVFH algorithm first create feature descriptors of geometric properties of two point clouds and then compare them. The reason we need CVFH in addition to ICP is that ICP is known to get stuck in local minima, which then it would falsely register two point clouds that are drastically different from each other. There-

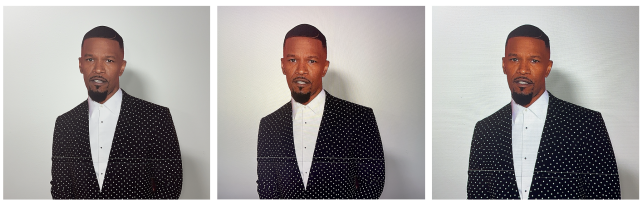


Figure 9: *Example from our dataset showing different display mediums used.*

fore, we need a fail-safe to verify the convergence result given by ICP. As CVFH tends to be computationally heavy, we first calculate the mean squared error (MSE) difference between two point clouds, and CVFH is only used if their it exceeds a set threshold. The logic behind this is that a lower MSE difference represents higher similarity and better alignment of two point clouds, and they are both positive indicators for ICP.

8 Dataset

Even though the Internet contains many fake/recaptured photos and videos, they do not contain depth information making them unsuitable for evaluating Scoop. Moreover, although there are existing RGBD datasets, to the best of our knowledge, none of them is constructed with recaptures in mind. We found that existing RGBD datasets consists of various 3D objects but have little or no recapture scenes, meaning that they cannot fully assess the capability of systems like Scoop. Therefore, in order to properly evaluate Scoop, we collect an RGBD dataset containing photos and videos captured with recaptures in mind. Our dataset covers a wide range of recapture scenarios. More specifically, we cover all flat display mediums (listed below) as well as a diverse set of content to be shown on them. We note that our main focus is on TV-based *full-recapture* attacks, which we believe to have the most imminent threat, but we also collected data of other recaptures with different flat display mediums and *partial-recapture* scenes so that we can thoroughly evaluate the potential of Scoop. On the other hand, we did not include any potential recapture attack that is defined as out of scope in §3, e.g., a misleading recapture of a flat surface/object shown on a display medium.

Display mediums. We use the following display mediums to collect our dataset:

1. Screen: a 85-inch Samsung The Frame LS03B LCD TV, which has a special coating for reducing reflection and tuned color and brightness mode for displaying photos like realistic arts, a 77-inch LG B4 OLED TV, and a 65-inch TCL R646 Mini-LED TV.
2. Projection: an Epson PowerLite L510U projector.
3. Printed: three life-size whole-body cardboard cutouts with different skin colors (two of them are celebrities and another one is an author of this paper) and a life-size celebrity face cutout.

Figure 9 demonstrates three example photos for each of the three options in the flat surface category: Figure 9(1) is a life-size cardboard cutout of a celebrity; Figure 9(2) is Figure 9(1) displayed on a TV; Figure 9(3) is Figure 9(1) displayed on a projector. Any of the three options may potentially fit into a malicious scenario; for example, they could be used to claim the presence of that celebrity at a certain location using the GPS location embedded in the provenance information [7] of those photos.

Our dataset also covers benign cases where display mediums are present, but their presence can be clearly recognized by the content viewer, as mentioned in §2.1 and shown in Figure 4.

Content. We collect some photos/videos of real environments to be used in the dataset. These contents are captured by a smartphone camera and are included in the dataset as benign scenarios. We also use the same photos/videos in the recapture scenarios. That is, we show them on our display mediums and capture photos/videos of the mediums in a way that it is not obvious that they are being displayed. For our dataset, we assign the ground truth to each photo/video to be either benign content or misleading recapture.

To thoroughly evaluate the effectiveness of our techniques, we try to collect a diverse content set. To do so, we use a total of three different categorization methods when collecting the content: background, object, and lighting.

Our categories for "background" include:

1. Plain (i.e., very simple color and/or depth distributions, such as white walls)
2. Textured (i.e., relatively simple color and/or depth distributions, such as patterned mats)
3. Complex (i.e., complex color and/or depth distributions, such as a shelf fully stacked with items)

Our categories for "object" include:

1. None (i.e., no specific object other than the background)
2. Item (i.e., item(s) is shown as the main object; the item could be small (e.g., a water bottle) or large (e.g., a TV))
3. Human/Animal (i.e., living being such as a person)
4. Mixed (i.e., mix of the above two)

Our categories for "lighting" include (note that all options can be either indoor or outdoor):

1. Very dark (i.e., from no light to the captured footage being barely visible)
2. Dark (i.e., most parts of the captured footage are visible, but only little light is presented)
3. Lit (i.e., the captured footage is clearly visible, but shadows can be easily found)

4. Well lit (i.e., the captured footage has almost no shadow)

Please refer to Appendix C for sample photos of different categories. We mix and match almost all options of the above three categories to the best of our effort in order to have the content set contain scenes as versatile as possible. We note that in some scenes, the object category's option could completely cover the background category's option, making the latter invisible.

8.1 Collection of Our Dataset

We first mix and match the normal three categories to capture different scenes, where we have multiple different settings for each type of scene, and there is one photo and one 10-second video clip for the data point of each setting. Followed by that, we then replay them using options in various display mediums. This results in a total of 488 data points collected in the final dataset, comprising 122 unique ones captured as photos and videos and on two different smartphones. We made use of two smartphones in order to learn the difference between different types of depth sensors (dToF vs. iToF); more specifically, we used two smartphones (Apple iPhone 14 Pro and Samsung Galaxy S20 Plus to capture each data point. More technical details about our dataset such as RGB and depth resolutions can be found in §9.

9 Prototype & Evaluation

We have implemented a complete prototype of Scoop. The producer (camera) side is implemented on both iOS and Android platforms. On the iOS platform, Scoop captures depth frames with a resolution of 768×432 during photo capture and 320×180 at 30 frames per second during video capture. On the Android platform, Scoop captures depth frames with a resolution of 310×205 during both photo capture and same resolution at 20 frames per second during video capture. On both producer platforms, their captured RGB photo has a resolution of 3840×2160 with JPEG codec, and same resolution for RGB video at 30 frames per second with H.265 codec. The iOS platform is evaluated on an Apple iPhone 14 Pro running iOS 18.2, and the Android platform is evaluated on a Samsung Galaxy S20 Plus running Android 13. It is noteworthy mentioning that the dToF sensor on the iPhone is pre-calibrated with the primary RGB camera sensor, where the iToF sensor on the Samsung phone is not. Such calibration is crucial for Scoop to work properly, as we rely on the fact that both depth and RGB camera sensors capture the exact same content. Although we manually calibrated the Samsung phone's iToF sensor with its primary RGB camera sensor, the lack of sensors' factory information make it hard for our calibration result to match with the iPhone. We used our producer prototype on both phones to capture our dataset, and iOS captured footage is used for playback on display mediums due to its higher picture quality.

The consumer (viewer) side has two components: generator and analyzer, where generator is responsible for producing the RGB frame based depth map using monocular depth estimation learning-based model and analyzer contains the algorithms for comparing the two depth maps for each RGB frame. The generator uses a state-of-the-art model: depth-pro [34], which utilizes PyTorch 2.5.1 [61] with CUDA 12.4 [2]. The analyzer is implemented with OpenCV 4.x (11-02-2024) [36] and PCL (10-27-2024) [65] libraries using C++. Both generator and analyzer run on top of Ubuntu 24.04.

The consumer uses a single CPU core and a GPU. We use the Intel Xeon Gold 6438M CPU (which has a lower single-thread performance than desktop CPUs) and an Nvidia RTX-4090 GPU, which is a consumer-grade GPU. Admittedly, this is a powerful GPU. We have not evaluated the performance of the consumer on weaker GPUs. But we note that Scoop's consumer could be performed off-line (e.g., in a server), the result of which could be simply appended as metadata to a photo/video (which then could be easily verified in any consumer device). We leave it to future work to optimize Scoop's consumer to run with good performance in various consumer devices.

Figure 10 show-cases Scoop viewer's prototype. We can clearly see that some facial features, details of clothes, hands, and gift boxes have depth information registered in the learning-based point cloud, which contradicts with the ground truth point cloud. In this case, we simply highlight all the violated parts with red color in the final displayed photo.

9.1 Effectiveness

The prototype of Scoop viewer and its parameters are developed using some test data we collected in our daily lives. Similar to the dataset we have, those test data covers a wide range of camera use cases, including capturing benign scenes and display mediums. We note that this set of test data is completely disjoint from our evaluation dataset. We then evaluate it on the dataset we collected, for both true positive rate (TPR) and false positive rate (FPR). True positive means successfully classifying the existing of a misleading recapture, where false positive means falsely classifying a benign scenario as a misleading recapture. Figure 11 shows the results on different display mediums; Figures 12(a-c) show the results on different benign content categories; and Figure 13 shows the overall results. In Figure 11, each of the first four x-axis options means the screen is captured without its frame being visible in photos, making them fall into *full-recapture* scenarios. The cutouts option represents a cutout put in front of either plain or textured background without any other objects, which is counted as a part of *partial-recapture* scenarios. Lastly, the mixed option indicates the photos contain either multiple cutouts or display mediums blended with real 3D object(s) , which also falls into *partial-recapture* attacks.

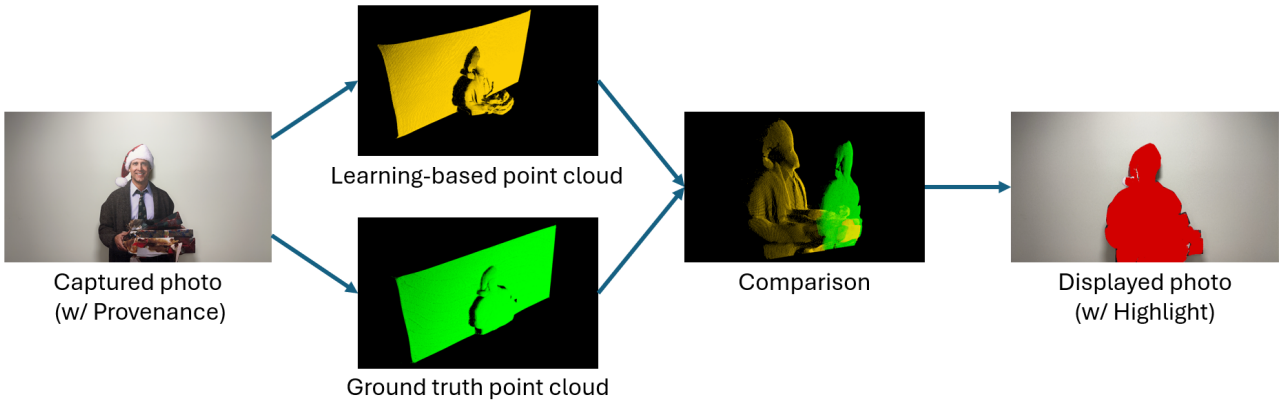


Figure 10: *Demonstration of Scoop viewer prototype's workflow with a sample photo containing a life size cardboard cutout of a celebrity.*

In summary, the iPhone 14 Pro (w/ dToF) based prototype achieves exceptional overall results with 94.81% of TPR and only 0.02% of FPR; The Galaxy S20 Plus (w/ iToF) based prototype achieves good overall results as well with 74.03% of TPR and 17.78% of FPR.

One observation is that the Android prototype struggles in complex scenes; for instance, the mixed option in Figure 11 points out that the Android prototype cannot even reach half the TPR as of what the iOS prototype is capable of achieving. Another observation about the Android prototype is it seems to be more sensitive to reflective surfaces; for example, the LG TV's screen is more reflective than other display mediums, and even if our data on that TV was captured in a very dark room, as shown in Figure 11, the Android prototype still does not perform well. We then further discovered that whenever we have scenes allowing lights to be bounced back and forth in different paths and distances, the Android prototype seems to perform worse. Such observation aligns with how the industry reviews dToF and iToF sensors: that dToF sensors generally have greater capability under complex lighting situation and long distance measurement and iToF sensors tend to suffer more from noise (e.g., multi-path interference) [8, 28, 44]. In addition to the fundamental technical difference, proper optimizations of hardware, firmware, and software (i.e., algorithms) matter as well. And our observation is that iPhone's dToF sensor is better calibrated.

Comparison with human's performance. In the user study we conducted, the participants have achieved 44.19% of TPR and 56.1% of FPR. In comparison, for the same set of photos, our iOS prototype achieves 100% of TPR and 12.5% of FPR, where the Android prototype achieves 87.5% of TPR and 12.5% of FPR. We believe that Scoop is more capable in terms of recognizing misleading recaptures than human beings.

To learn about the effect of different depth resolution, we have also conducted another round of experiments on the same iOS captured content, but with downscaled depth resolution (320×180), which roughly matches with the Android

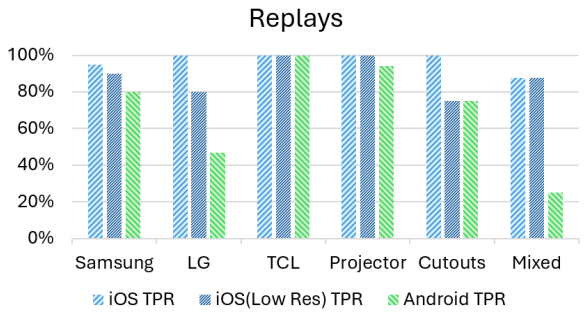


Figure 11: *Results of Scoop's prototype under different display mediums.*

counterpart. As seen in Figure 13, at the lower depth resolution, iOS prototype still manages to retain a great TPR result of 90.91%, but suffers from a worsen FPR of 17.78%. We noticed that in benign scenes with lower depth resolutions, whenever most of the captured scene is filled with flat surfaces, it is more likely that the ground truth depth map would disagree with the learning-based depth map as learning-based depth maps tend to contain more subtle details in these scenes, and this is proven by the relatively high false positive rates of both the iOS prototype with lower depth resolution and the Android prototype on None and Item options in Figure 12(b).

9.2 Overheads

Performance. Scoop introduces computational overhead in three main places: camera capture, learning-based depth estimation, and viewer analysis. At camera capture time, although additional computation is performed, no human noticeable runtime overhead has been observed in both photo and video capturing. As Figure 14(a) shows, depth estimation takes on average of 0.28 ±0.02 second for the iOS prototype captured photo (no matter what the depth resolution is) and 0.26 ±0.01 second for the Android prototype captured photo; Scoop viewer's analysis, on the other hand, introduces the most runtime overhead: an average of 69.38 ±95.17 seconds

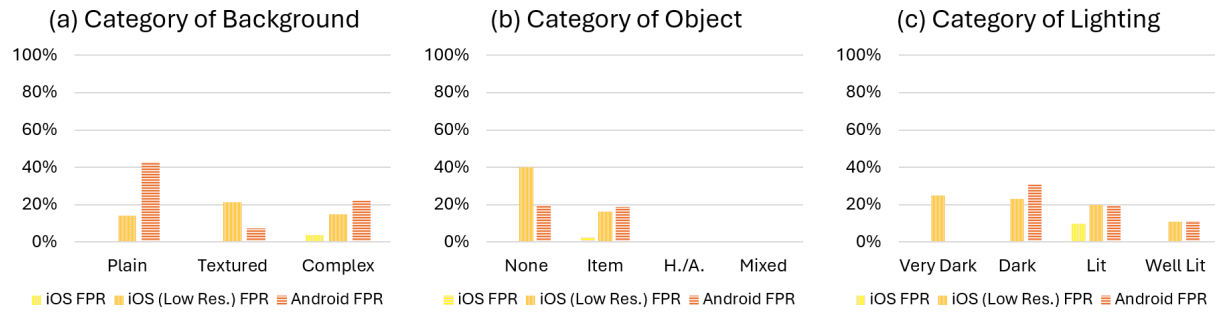


Figure 12: *Results of Scoop's prototype under different benign content categories.*

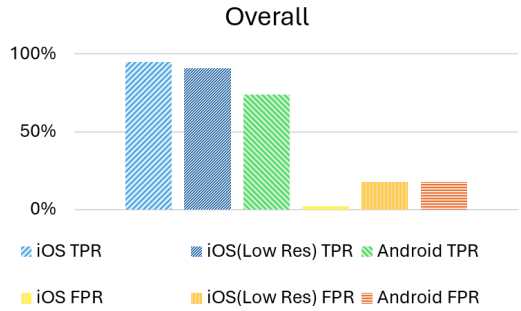


Figure 13: *Overall effectiveness of Scoop's prototype.*

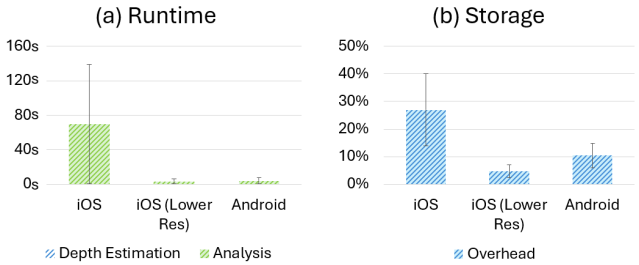


Figure 14: *Both performance and storage overheads of Scoop's prototype.*

for the iOS prototype, 2.96 ±3.48 seconds for the iOS prototype with lower depth resolution, and 3.68 ±2.99 seconds for the Android prototype. Note that the analysis duration varies significantly due to difference in scene complexity and usage of our techniques, resulting in large standard deviations. While this statistical variation produces negative values, all actual measured durations in our experiments were positive. The reason our depth estimation runs much faster is it uses well-optimized software (i.e., PyTorch with CUDA support) and powerful hardware (i.e., RTX 4090); in contrast, our viewer analysis runs slow, as it is currently not optimized for performance. We also note that standard deviation of analysis' runtime for both platforms are large. This is due to our techniques (§7.2), where sometimes running only one technique is enough, and other times all techniques are needed. We can also observe that the content captured by the iOS platform has worse runtime overhead than the Android prototype. This is caused by both the fact that the iOS prototype capturing depth maps with higher resolution than the Android counterpart and our picks of point cloud algorithms where some of them have runtime complexities worse than linear.

An alternative approach to boost performance is to perform frame-level comparison instead of region-level comparison. As only one comparison needs to be done in this way, the overhead can be significantly reduced and such light version of Scoop is still capable of detecting misleading *full-recaptures*, but not *partial-recaptures*.

Storage. Figure 14(b) shows the percentage of extra storage space Scoop takes for each photo. The overhead on the iOS prototype is 648 KB (about 26.91 ±13.09%) for higher depth resolution and 112.5 KB (about 4.69 ±2.27%) for lower depth resolution. The overhead on the Android prototype is 125 KB (about 10.4 ±4.47%). As can be seen, the depth map size stays unchanged across photos, whereas photo sizes vary depending on the complexity of scenes (which determines how much compression can be done) and compression ratios determined by different software platforms (iOS v.s. Android).

Energy consumption. We use the Samsung Galaxy S20 Plus for this experiment, as there is no direct way to get detailed power consumption data on the iOS platform. By leveraging Android *Power Profiler* [14], we managed to measure the energy consumption of both capturing a RGB photo and a 10-second video using Scoop prototype camera. We also introduce a baseline application that functions the same but do not capture depth information. During photo capture, the baseline consumes energy at a rate of 1986.67 ±48.17 microamps and the Scoop prototype consumes energy at a rate of 2793 ±98.85 microamps. As it takes about 0.09 ±0.01 second to capture a RGB photo and 0.1 ±0.01 second to capture a RGB photo and a depth map, our prototype captures photos with an energy consumption overhead of 56.2%. During video capture, the baseline consumes energy at a rate of 1764.1 ±152.08 microamps and the Scoop prototype consumes energy at a rate of 2435.7 ±193.04 microamps. Since there is no difference in capture times for both of them (10s for videos), we conclude that our prototype incurs an energy consumption overhead of 38.07% during video capture.

10 Additional Considerations

10.1 3D Display-Based Attacks

In Scoop, we consider attacks using 3D display mediums to be out of scope. For these attacks, we assume custom-shaped (e.g., a 3D-printed model) objects are used along with display mediums (e.g., 3D projection). To some extent, we do not think our system would fail easily against naive versions of such an attack. As explained in 7, Scoop works by applying regional analysis based on absolute depth information, so in order to spoof Scoop, the 3D objects need to be refined enough to surpass our comparison thresholds. Furthermore, if captured objects are not made in real life-size (e.g., miniatures), Scoop can also utilize triangulation to figure out the difference in sizes between the ground truth depth map and the learning-based depth map. While we do not perform any direct size comparison in our current prototype, we note that Scoop's Technique (3) (as mentioned in §7.2) is likely able to handle this due to the use of rigid transformation without any uniform scaling, where a large size difference will cause the convergence score to exceed our set threshold. In other words, for such an attack to succeed, it has to be mounted with life-size refined custom shapes along with high quality display mediums. That said, we opt to not claim any capability against 3D display-based attacks without a comprehensive evaluation, which will be a future work of Scoop. Moreover, we believe that flat display mediums are the immediate threat to provenance-based media authentication, and this work's main contribution is about tackling flat display-based recapture attacks.

10.2 Binocular Vision

In this work, we explicitly choose to make use of ToF-based depth sensors, as they are widely used in the smartphone industry. We considered binocular vision technology [15], which works similarly to human eyes by making use of two camera sensors in a device to measure the absolute distance between the device and captured objects. However, such devices are rare and relatively new [17, 18]. It is unclear how well they work for distance estimation, and particularly estimating the relative distance between objects and their depth. It is possible that their use, in addition to ToF, could make Scoop more robust.

10.3 Forward Looking Infrared (FLIR)

Some recent phones, such as the Google Pixel 9, are equipped with a sensor [24] that can measure the temperature of distant surfaces. Some prior work uses temperature to generate depth maps [10, 66], which works exceptionally well on scenes containing human bodies, and we acknowledge that this is still an ongoing research topic. However, they are mostly limited to known objects with relatively fixed ranges of temperature - knowingly human beings. To make matters worse, such sensors on existing smartphones have limited resolution for distant objects. There is still potential light in this path though, military-grade FLIR cameras can produce 720p resolution heat maps for up to 10 kilometers or 6 miles [22]. We hope that over time, the price point and energy consumption of those sensors can be brought down to meet the requirements to be included in smartphones.

10.4 Videos

As videos are essentially bundles of photos (frames) being play-backed in sequence, Scoop supports videos out of the box. However, its support of videos is also very preliminary and can be vastly improved. Here we list three points that we believe can be used to enhance Scoop for videos, but we do note that more can be done. First, consecutive frames in videos tend to contain continuous motion, especially in videos captured with a camera; with the support of either software algorithms and/or hardware such as gyroscope and accelerometer, point clouds can be built across multiple frames for higher accuracy. Second, as frames are being play-backed at high speed, where each frame has limited visible time, assigning suspiciousness on a frame-by-frame based is most likely not needed. Third, similar to the second point, depth maps may not need to be captured on a per-frame based.

Although various enhancements can be applied to videos, it is noteworthy to mention that the adversary may also take advantage of unique features of videos. For example, they may move the camera and display medium freely at capture time to craft an illusion of movement. Another dangerous factor is that videos have more post-processing potentials. Therefore, we believe that video-specific enhancements should be considered thoroughly before being applied to Scoop.

10.5 Post-Processing

The captured content may need further post-processing, such as cropping, coloring, or applying other filters. Similar to the provenance information proposed by Vronicle [57], the embedded depth information needs to get updated along with its corresponding RGB photo. This does not mean that the depth information can be freely updated whenever a change is applied to its RGB counterpart, as this could easily open up an attack vector for the adversary. Instead, the depth information can only be applied with subtraction-based update. This means that every time an update is being applied, the update must be a removal of depth information. For example, when the RGB photo is cropped, its depth information is also cropped with the same scale. Another example would be when a person's face is blurred in the RGB photo, the face's depth information is also removed (by setting 0 to all face-related pixels). On the other hand, if an object is artificially added to

the RGB photo, no depth of that object is applied to the depth information of the original photo; instead, all pixels covered by the newly added object would have their depth counterparts zeroed out, effectively turning the covered area into a flat surface with no depth value, and this can be picked up by Scoop viewer. During post-processing, all updates to the depth information are done automatically according to how the corresponding RGB pixels are modified and users (editors) are not allowed to directly make any change to the depth information, where all of these are enforced by provenance-based secure post-processing systems such as Vronicle [57].

11 Related Work

Provenance-based authentication. As photo and video modification tools have become increasingly sophisticated, especially with the advent of generative AI, content authenticity is increasingly important. To this end, various methods were developed to restore credibility of visual content [7, 48, 54, 55, 57, 58]. One promising approach is provenance-based authentication [7, 57, 58], since it provides traceability of digital content with a verifiable cryptographic proof. Many major industry players (including BBC, Adobe, and Microsoft [5]) either already adopted provenance-based countermeasures or announced plans to do so in the near future. Scoop augments provenance-based solutions by providing depth information in the provenance metadata, allowing for accurate discovery of misleading recaptures.

Display screen recapture detection. There has been quite some research effort put on detecting recaptured images that were displayed on a screen, where most of them have been focusing on detection of LCD screen displayed images [37, 50, 72]. They rely on the unique moiré pattern [20] that could be detected on LCD displays. There is also work that further extends this technique to OLED display [74]. Most of those works utilize machine learning to help recognizing the almost invisible unique patterns of each display, where the models need to be trained on the exact display model's panel. Given the fast development of display technology, rapid release, and extensive amount of existing displays in the market, it is hard to come up with a model that is capable of detecting all display models. Furthermore, there are more kinds of display methods that can be used to display arbitrary photos and videos, such as projecting, printing, and so on.

Face anti-spoofing. There are many previous works that focus specifically on face recapture detection [13, 43, 52, 56]. This is due to the fact that face is widely used as an authentication method nowadays, where deceiving the facial authentication system may bring enormous benefits to the adversary. Some commercial solutions [11, 27] already exist. They use additional sensors to help detecting displayed images by either filtering unwanted display surfaces (e.g., printed photos or display screens) or reconstructing 3D model of the user's face. There is also research work [43] done for attempting to reconstruct user's 3D face model without the use of additional sensor. Other works [52, 56] utilize facial feature extraction and analysis with machine learning model to figure out the difference between real faces and displayed faces. Another line of work [13] is to conduct active liveness detection, which requires the user to perform randomly assigned actions in front of the camera. Although all these above works are reasonably powerful, they are highly limited to only one specific use case - face authentication, where Scoop aims to be a more generic approach for both photos and videos.

Object anti-spoofing. Serval previous works [41, 51, 53, 69] have considered object anti-spoofing on top of face anti-spoofing, where most of them have all opted for machine learning based approaches mToFNet [51] also makes use of the ToF sensor presented on the camera, but only as an auxiliary input to the model. Additionally, these solutions are limited to either the type of displays that their models are trained on or the kind of objects that their models are optimized for.

12 Conclusions

Visual content is an essential form of information consumption by humans. A variety of consequential news is obtained in this medium, including presidential addresses, parliamentary debate, police action, and war footage. We now increasingly rely on such content for automation of critical tasks, including driving, manufacturing, and border control. Provenance-based techniques from both academia and industry protect the pipeline from camera sensor to consumption display, but fail to protect us from recapture attacks in front of the camera sensor. We presented Scoop, a solution designed to protect us from an adversary launching recapture attacks, where the physical scene is manipulated with digital screens or posters. We showed that it achieves high accuracy in finding misleading recaptures, which it presents to viewers to help them detect attacks. We hope our work will inspire adoption of ToF sensors by more cameras, such as security cameras. We expect that the cost, range, and resolution of ToF sensors will improve as adoption increases. We expect that with increasing use of multi-modal AI, even more sophisticated techniques will be built to interpret the depth information collected by Scoop. All these trends will contribute to making Scoop more powerful against such attackers.

13 Acknowledgment

The work was supported in part by UCI ICS Exploration Research Award, NSF Award SATC1956393, and NSA Award H98230-22-1-0308. The authors thank the anonymous reviewers for their insightful comments. They also thank the shepherd for providing significant help with preparing a revised and improved version of the paper.

14 Ethics Considerations

This paper is motivated by recapture attacks that pose increasing threats to society's trust in visual content. We constructed Scoop, a systematic mitigation for some of these attacks. We also collected a dataset containing many data points that were used to thoroughly evaluate our approach. Moreover, we conducted a user study to show that human beings cannot distinguish between original and recaptured photos, where Scoop can easily outperform them.

Beneficence. Our intent is for this work to yield better public awareness of the dangers of recapture attacks. We also provide a means of mitigating some (though not all) types of such attacks. One potentially negative impact of this work is the possibility of someone being motivated by this paper to conduct such attacks. This is further discussed in "Justice" below.

Respect for Persons When developing and evaluating Scoop prototype, we collected and used some test data that involved human participation. This was limited to the authors of this paper, each of whom has given explicit consent for their photos and videos (no audio) to be collected for that purpose. Also, all collected test data was used privately and immediately deleted thereafter, with no backup whatsoever. In addition, the collected dataset contains human images that also involve only the authors, each of whom consented for their photos and videos (no audio) to be collected and potentially published after the paper is published.

The user study had been conducted with full disclosure of the process to the participants prior to their participation. Furthermore, there was no personally identifiable information (PII) collected or stored throughout the user study, where none of the responses can be linked to any individual. Each participant was also treated respectfully and compensated with the same amount of rewards (i.e., a $10 Amazon gift card) upon completion of each user study session, where their performance (i.e., whether they can correctly classify photos or not) did not impact their compensation.

Besides the above, this work did not collect, use, or potentially endanger any person.

Justice As mentioned above (in Beneficence), we acknowledge that this work raises awareness of recapture attacks, which – though generally a positive outcome – might trigger someone to try mounting such attacks in practice. We believe that public awareness of recapture attacks is just a matter of time. Thus, we consider that earlier awareness is beneficial, especially since this work provides a practical and effective means of mitigating some such attacks. This gives us confidence in this paper being pro-justice, rather than anti-justice.

Respect for Law and Public Interest We made sure to obey all relevant laws, especially during the development of Scoop and collection of the dataset. For example, since our work entailed human involvement and collecting personally identifiable information (authors' faces), we followed all guidelines provided at the authors' institutions and submitted an IRB application for the public release of the data, which was determined to be an exempted case, meaning that no review was needed. Some domestic pets (cats) were also involved in evaluating Scoop prototype and in dataset collection. However, these pets were/are owned by authors, and no harm was caused and no forced behavior was involved. A separate IRB application for the user study was also submitted, which was also determined to be an exempted case. Note that we made both determinations by using the institute-provided protocols and tools.

15 Open Science

This paper is in full compliance with the Open Science Policy. It yields two artifacts: Scoop research prototype and the collected dataset. As stated earlier, we will fully open-source the prototype once the paper is published. Also, we intend to publicly release the full dataset once the paper is published and the IRB application is approved. Recall that the dataset needs IRB approval since it contains personal identifiable information of the authors; this is also discussed in §14.

References

[1] Distorted Videos of Nancy Pelosi Spread on Facebook and Twitter, Helped by Trump. https://www.nytimes.com/2019/05/24/us/politics/pelosi-doctored-video.html.

[2] NVIDIA CUDA. http://www.nvidia.com/object/cuda_home_new.html.

[3] Truepic Breakthrough Charts a Path for Restoring Trust in Photos and Videos at Internet Scale. https://www.prnewswire.com/news-releases/truepic-breakthrough-charts-a-path-for-restoring-trust-in-photos-and-videos-at-internet-scale-301152998.html.

[4] QuestionPro: Online Survey Software and Tools. https://www.questionpro.com/, 2002.

[5] Content Authenticity Initiative. https://contentauthenticity.org/, 2019.

[6] Check out the new camera functions of the Galaxy S20 | S20+ | S20 Ultra. https://www.samsung.com/uk/support/mobile-devices/check-out-the-new-camera-functions-of-the-galaxy-s20-plus-s20-ultra, 2020.

[7] Coalition for Content Provenance and Authenticity. https://c2pa.org/, 2021.

[8] Comparison between dToF and iToF sensors. https://faster-than-light.net/TOFSystem_C4, 2021.

[9] Dutch MPs in video conference with deep fake imitation of Navalny's Chief of Staff. https://nltimes.nl/2021/04/24/dutch-mps-video-conference-deep-fake-imitation-navalnys-chief-staff, 2021.

[10] Heat Map Spectrum to Grayscale (Depth Maps). https://github.com/ImageMagick/ImageMagick/discussions/3126, 2021.

[11] Windows Hello face authentication. https://learn.microsoft.com/en-us/windows-hardware/design/device-experiences/windows-hello-face-authentication, 2021.

[12] iPhone 14 Pro. https://support.apple.com/guide/iphone/iphone-14-pro-iph6928b4ea3/ios, 2022.

[13] FaceRecognition-LivenessDetection-Android. https://github.com/MiniAiLive/FaceRecognition-LivenessDetection-Android, 2023.

[14] Power Profiler. https://developer.android.com/studio/profile/power-profiler, 2023.

[15] Stereo Vision and Depth Perception in Computer Vision. https://sumitkrsharma-ai.medium.com/stereo-vision-and-depth-perception-in-computer-vision-8aff8dd04e7c, 2023.

[16] Adobe Content Authenticity. https://contentauthenticity.adobe.com, 2024.

[17] Apple Vision Pro gets extraordinary $30,000 dual-lens camera that can handle almost 118 million pixels. https://www.techradar.com/pro/apple-vision-pro-gets-extraordinary-usd30-000-dual-lens-camera-that-can-handle-almost-118-million-pixels, 2024.

[18] Canon Now Accepting Orders for Spatial Video Lens Previewed at WWDC. https://www.macrumors.com/2024/11/03/canon-spatial-video-lens-pre-orders/, 2024.

[19] Iterative Closest Point Algorithm. https://cs.gmu.edu/~kosecka/cs685/cs685-icp.pdf, 2024.

[20] moiré pattern. https://www.britannica.com/science/moire-pattern, 2024.

[21] Moondream AI. https://moondream.ai/, 2024.

[22] Ranger® HDC. https://www.flir.eu/products/ranger-hdc/, 2024.

[23] Sora. https://openai.com/index/sora/, 2024.

[24] Take the temperature of objects with your Pixel phone. https://support.google.com/pixelphone/answer/14103759, 2024.

[25] Time-of-Flight principle. https://www.terabee.com/time-of-flight-principle, 2024.

[26] Trust what you see: How Truepic authenticates images and videos in the age of deepfakes. https://www.qualcomm.com/news/onq/2024/11/trust-what-you-see-how-truepic-authenticates-images-and-videos, 2024.

[27] Use Face ID on your iPhone or iPad Pro. https://support.apple.com/en-us/108411, 2024.

[28] What is LiDAR? https://www.symphotony.com/lidar/principle/, 2024.

[29] Zenmuse L2 - DJI. https://enterprise.dji.com/zenmuse-l2, 2024.

[30] Gemini AI video generation. https://gemini.google/overview/video-generation, 2025.

[31] AVSYSTEM. ToF: Time-of-Flight - Overview, Principles, Advantages. https://avsystem.com/blog/linkyfi/time-of-flight, 2024.

[32] BHAT, S., BIRKL, R., WOFK, D., WONKA, P., AND MÜLLER, M. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288* (2023).

[33] BIRKL, R., WOFK, D., AND MÜLLER, M. Midas v3.1 – a model zoo for robust monocular relative depth estimation. *arXiv preprint arXiv:2307.14460* (2023).

[34] BOCHKOVSKII, A., DELAUNOY, A., GERMAIN, H., SANTOS, M., ZHOU, Y., RICHTER, S., AND KOLTUN, V. Depth pro: Sharp monocular metric depth in less than a second. *arXiv* (2024).

[35] BOULKENAFET, Z., KOMULAINEN, J., LI, L., FENG, X., AND HADID, A. Oulu-npu: A mobile face presentation attack database with real-world variations. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)* (2017).

[36] BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[37] CAO, H., AND KOT, A. Identification of recaptured photographs on lcd screens. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing* (2010).

[38] CENTER, P. R. How Americans Get News on TikTok, X, Facebook and Instagram. HowAmericansGetNewsonTikTok,X,FacebookandInstagram, 2024.

[39] CENTER, P. R. Social Media and News Fact Sheet. https://www.pewresearch.org/journalism/fact-sheet/social-media-and-news-fact-sheet/, 2024.

[40] COLE, S. Hacked News Channel and Deepfake of Zelenskyy Surrendering Is Causing Chaos Online. https://www.vice.com/en/article/93bmda/hacked-news-channel-and-deepfake-of-zelenskyy-surrendering-is-causing-chaos-online.

[41] COSTA, V., SOUSA, A., AND REIS, A. Image-based object spoofing detection. In *Combinatorial Image Analysis* (2018).

[42] FARID, H. Image forgery detection. *IEEE Signal Processing Magazine* (2009).

[43] FARRUKH, H., ABURAS, R., CAO, S., AND WANG, H. Facerevelio: a face liveness detection system for smartphones with a single front camera. MobiCom '20.

[44] FENG, S. Understanding Principle of Time of Flight Technology. https://industry.goermicro.com/blog/tech-briefs/understanding-the-principle-of-3d-time-of-flight-camera.html, 2024.

[45] GILBERT, P., JUNG, J., LEE, K., QIN, H., SHARKEY, D., SHETH, A., AND COX, L. P. YouProve: Authenticity and Fidelity in Mobile Sensing. In *Proc. ACM SenSys* (2011).

[46] GODARD, C., MAC AODHA, O., FIRMAN, M., AND BROSTOW, G. Digging into self-supervised monocular depth prediction.

[47] HAN, X., FENG, Z., SUN, S., AND XIAO, G. 3d point cloud descriptors: state-of-the-art. *Artif. Intell. Rev.* (2023).

[48] HASAN, H. R., AND SALAH, K. Combating Deepfake Videos Using Blockchain and Smart Contracts. *IEEE Access 7* (2019), 41596–41606.

[49] HU, M., YIN, W., ZHANG, C., CAI, Z., LONG, X., CHEN, H., WANG, K., YU, G., SHEN, C., AND SHEN, S. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation.

[50] HUSSAIN, I., HUSSAIN, D., KOHLI, R., ISMAIL, M., HUSSAIN, S., SAJID ULLAH, S., ALROOBAEA, R., ALI, W., AND UMAR, F. Evaluation of deep learning and conventional approaches for image recaptured detection in multimedia forensics. *Mobile Information Systems* (2022).

[51] JEONG, Y., KIM, D., LEE, J., HONG, M., HWANG, S., AND CHOI, J. mtofnet: Object anti-spoofing with mobile time-of-flight data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (2022).

[52] LI, H., LI, W., CAO, H., WANG, S., HUANG, F., AND KOT, A. Unsupervised domain adaptation for face anti-spoofing. *IEEE Transactions on Information Forensics and Security* (2018).

[53] LI, J., KONG, C., WANG, S., AND LI, H. Two-branch multi-scale deep neural network for generalized document recapture attack detection. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2023).

[54] LI, Y., CHANG, M., AND LYU, S. Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking. In *Proc. IEEE International Workshop on Information Forensics and Security (WIFS)* (2018).

[55] LI, Y., AND LYU, S. Exposing Deepfake Videos by Detecting Face Warping Artifacts. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2019).

[56] Liu, Y., Jourabloo, A., and Liu, X. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018).

[57] Liu, Y., Nakatsuka, Y., Amiri Sani, A., Agarwal, S., and Tsudik, G. Vronicle: Verifiable provenance for videos from mobile devices. MobiSys '22.

[58] Liu, Y., Nakatsuka, Y., Amiri Sani, A., Agarwal, S., and Tsudik, G. Provcam: A camera module with self-contained tcb for producing verifiable videos. ACM MobiCom '24.

[59] Matyás, V., and Říha, Z. Biometric authentication - security and usability. In *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security* (2002).

[60] Naveh, A., and Tromer, E. PhotoProof: Cryptographic Image Authentication for Any Set of Permissible Transformations. In *Proc. IEEE Symposium on Security and Privacy (S&P)* (2016).

[61] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library, 2019.

[62] Piccinelli, L., Yang, Y., Sakaridis, C., Segu, M., Li, S., Van Gool, L., and Yu, F. UniDepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024).

[63] Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., and Koltun, V. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[64] Rustamov, A., Gogoi, N., Minetto, A., and Dovis, F. Assessment of the vulnerability to spoofing attacks of gnss receivers integrated in consumer devices. *2020 International Conference on Localization and GNSS (ICL-GNSS)* (2020), 1–6.

[65] Rusu, R., and Cousins, S. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)* (2011), IEEE.

[66] Sarandi, I., Linder, T., Arras, K., and Leibe, B. Metrabs: Metric-scale truncation-robust heatmaps for absolute 3d human pose estimation. *IEEE Transactions on Biometrics, Behavior, and Identity Science* (2021).

[67] Satariano, A., and Mozur, P. The People Onscreen Are Fake. The Disinformation Is Real. https://www.nytimes.com/2023/02/07/technology/artificial-intelligence-training-deepfake.html.

[68] Statt, N. Thieves are now using AI deepfakes to trick companies into sending them money. https://www.theverge.com/2019/9/5/20851248/deepfakes-ai-fake-audio-phone-calls-thieves-trick-companies-stealing-money, 2019.

[69] Stehouwer, J., Jourabloo, A., Liu, Y., and Liu, X. Noise modeling, synthesis and classification for generic object anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[70] Stocking, G., Van Kessel, P., Barthel, M., Eva Matsa, K., and Khuzam, M. Many Americans Get News on YouTube, Where News Organizations and Independent Producers Thrive Side by Side. https://www.pewresearch.org/journalism/2020/09/28/many-americans-get-news-on-youtube-where-news-organizations-and-independent-producers-thrive-side-by-side/, 2020.

[71] Thomson, T., Angus, D., and Dootson, P. 3.2 billion images and 720,000 hours of video are shared online daily. Can you sort real from fake? https://theconversation.com/3-2-billion-images-and-720-000-hours-of-video-are-shared-online-daily-can-you-sort-real-from-fake-148630, 2020.

[72] Thongkamwitoon, T., Muammar, H., and Dragotti, P. An image recapture detection algorithm based on learning dictionaries of edge profiles. *IEEE Transactions on Information Forensics and Security* (2015).

[73] Toews, R. Deepfakes Are Going To Wreak Havoc On Society. We Are Not Prepared. https://www.forbes.com/sites/robtoews/2020/05/25/deepfakes-are-going-to-wreak-havoc-on-society-we-are-not-prepared/, 2020.

[74] Trabelsi, A., Pic, M., and Dugelay, J. Recapture detection to fight deep identity theft. VSIP '22.

[75] Wang, K. Time and position spoofing with open source projects.

[76] Westerlund, M. The Emergence of Deepfake Technology: A Review. http://doi.org/10.22215/timreview/1282.

[77] Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., and Zhao, H. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR* (2024).

[78] Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., and Zhao, H. Depth anything v2. *arXiv:2406.09414* (2024).

[79] Yin, W., Zhang, C., Chen, H., Cai, Z., Yu, G., Wang, K., Chen, X., and Shen, C. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023).

[80] Zhang, Z., Zhan, X., and Xu, H. Development and validation of a low-cost gps spoofing simulator. *Journal of Aeronautics, Astronautics and Aviation* (2014).

A Answer to The Figure

In Figure 1, the right one is a real photo. The left photo was displayed on our TCL Mini-LED TV, with the person in the photo digitally erased.

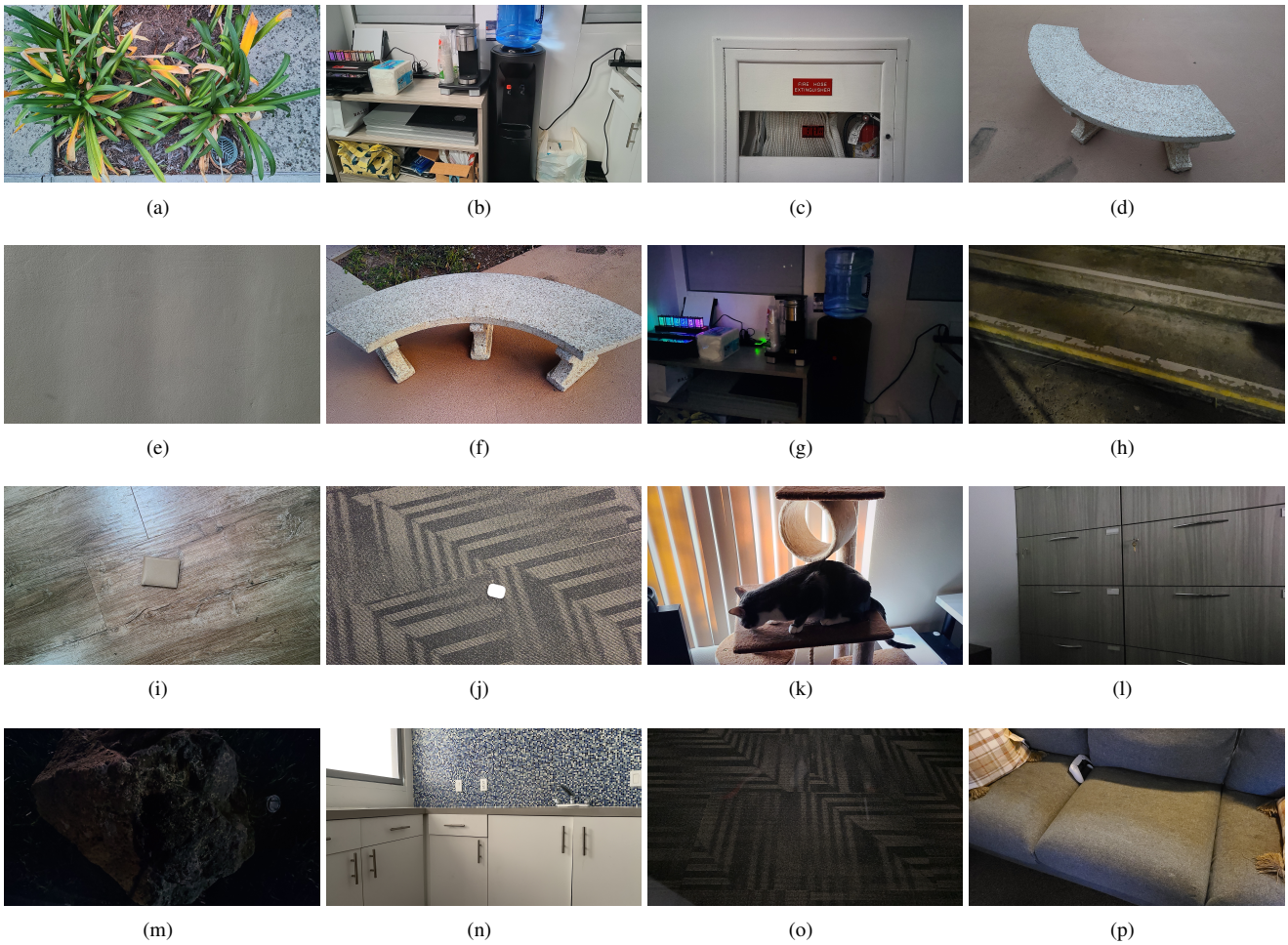
B User Study

B.1 Photos

Figure 15 shows all the photos we used in our user study. Original photos include (d), (e), (h), (j), (l), (n), (o), and (p). Recaptured photos include (a), (b), (c), (f), (g), (i), (k), and (m).

C Dataset Photos

Figure 16 shows an example for each option of the three categories as mentioned in the main paper. In addition to the sample photos for different categories in the dataset and user study, Figure 17 shows more photos from our dataset. Original photos include (a), (b), (c), and (d). Recaptured photos include (e), (f), and (g).



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

Figure 15: *Photos used in our user study.*

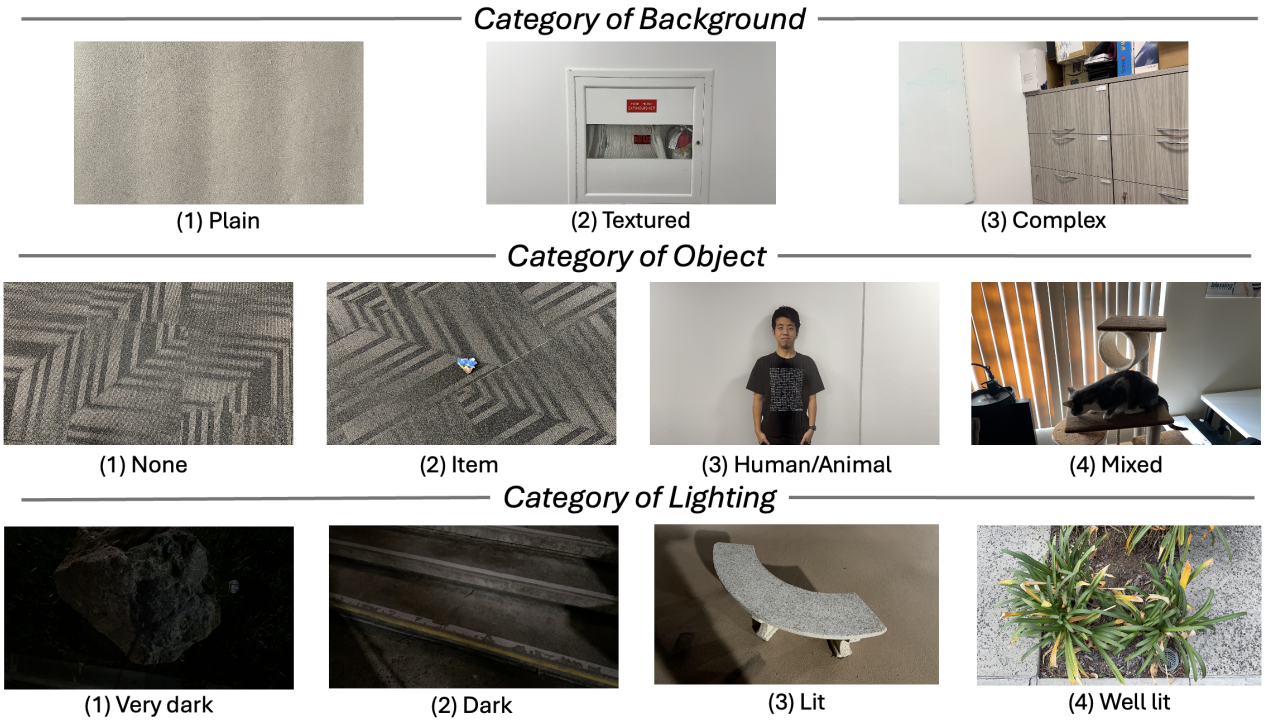


Figure 16: *Our dataset's three categories: background, object, and lighting.*

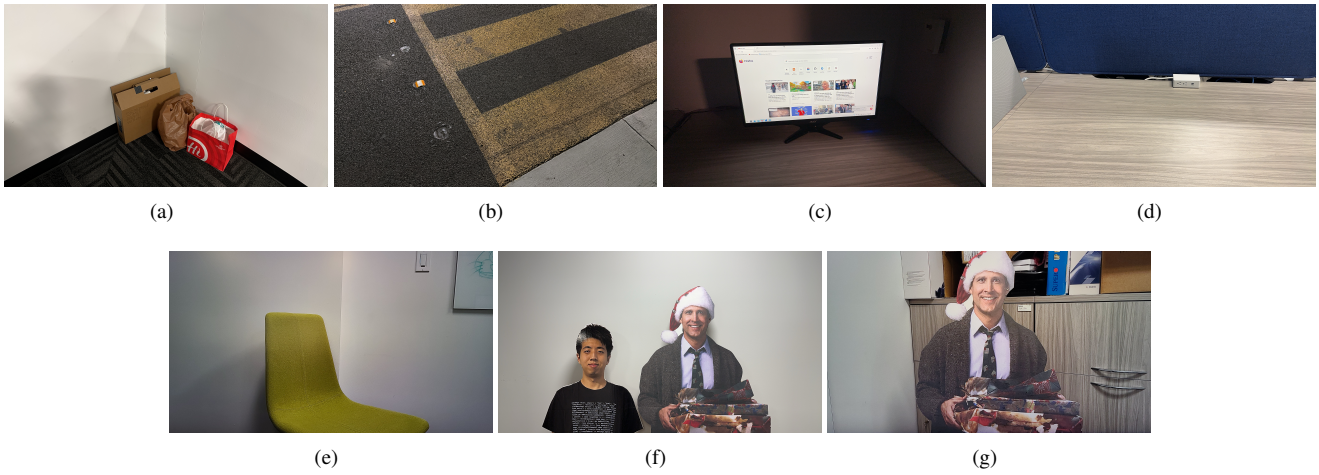


Figure 17: *More sample photos in our dataset.*

D Artifact

There are four artifact components in our paper.

- **Viewer**: The Scoop Viewer is the main tool that we developed to detect the existence of misleading recaptures in photos/videos.
- **iOS App**: The iOS App is a mobile application that can be used to capture photos/videos that can be analyzed by the Scoop Viewer.
- **Android App**: The Android App is a mobile application that can be used to capture photos/videos that can be analyzed by the Scoop Viewer.
- **Dataset**: The dataset that can be used to evaluate systems such as Scoop.

They can all be found on our Zenodo page at https://doi.org/10.5281/zenodo.15611905. There are three ways to test the Scoop system, which are described in the following subsections.

D.1 Quick Test (Recommended)

We provide a quick way to test out the Scoop system. Please download the Scoop Viewer (viewer.zip), which contains everything needed to conduct a quick test of Scoop.

The Scoop Viewer requires the following dependencies:

- C++ compiler (e.g., g++, clang++)
- CMake 3.5 or later
- OpenCV 4.5 or later
- PCL (Point Cloud Library) 1.12 or later
- Boost 1.75 or later
- Eigen 3.3 or later
- Python 3.9 or later

You may set up the dependencies yourself and modify the CMakeLists.txt file accordingly, or you can use our provided script to set up the environment automatically. Our script is tested on Ubuntu 24.04 LTS, but it should work on other Linux distributions (e.g., Fedora, Arch Linux) and MacOS (with Homebrew) as well. Assuming the viewer is now at a directory called `$VIEWER_DIR`, you can run the following commands to set up the environment:

```
cd $VIEWER_DIR
bash ./scripts/install_required_libraries.sh
```

After the installation is complete, you can run the following commands to build the Scoop Viewer:

```
cd $VIEWER_DIR
cmake .
make -j$(nproc)
```

After the build is complete, you can run the quick test with the following command:

```
cd $VIEWER_DIR
python3 ./scripts/eval.py sample_data/ 0000 9999
```

This command will run the Scoop Viewer on the sample data in the `sample_data` directory, which contains 8 sets of data provided (with 4 unique data points). Among the 4 data points, 1 is original and the rest 3 are recaptured photos (with 2 TVs and 1 projector).

D.2 Test with Your Own Data

You can also test the Scoop Viewer with your own data. To do this, you can either use our iOS or Android app to capture photos/videos, or you can use your own camera to capture photos/videos. However, please make sure you follow the guidelines in the iOS/Android app repositories for correctly capturing the data or refer to the Scoop Viewer repository for the correct format of the data. For building iOS and Android apps, please refer to the respective repositories for instructions.

After you have extracted the data from your iOS/Android app, you need to generate perceived depth data for each photo/video. Please refer to the Scoop Viewer repository for instructions on how to generate perceived depth data. You can pick any depth estimation model that you prefer, but we recommend using the ml-depth-pro model, which we included a script for using it in the Scoop Viewer repository. You may need to refer to the script to see how to generate the perceived depth data for your photos/videos and make sure the generated data is in the correct format so that the Scoop Viewer can analyze it. After you have generated the perceived depth data, you can run the Scoop Viewer on your data with the commands provided in the Scoop Viewer repository.

D.3 Test with the Dataset

You can also test the Scoop Viewer with our dataset. We provide full instructions on how to use the dataset in the Scoop Viewer repository. You can also refer to the Scoop dataset repository for more information about the dataset.