International Journal of Emerging Multidisciplinary Research And Innovation (IJEMRI)

Adversarial Image Integrity and Tamper Check Browser Extension

[1]T C Swetha Priya, [2]Qurrath Ul Aain, [3]Sayera Mehreen Khan

[1]Assistant Professor Dept. of Information Technology Stanley College of Engineering and Technology for Women Hyderabad, India tcswethapriya@stanley.edu.in
[2]Dept. of Information Technology Stanley College of Engineering and Technology for Women Hyderabad, India ulaainqurrath@gmail.com
[3]Dept. of Information Technology Stanley College of Engineering and Technology for Women Hyderabad, India sayeramahreenkhan@gmail.com

ABSTRACT

Client-side filtering tools typically protect user privacy through local transformations of media files. However, existing tools depend on third-party trust or operate outside of the daily upload workflow. We developed a Chrome Extension that leverages Five-Layer Adversarial Pixel Perturbation and casts transformations on user images directly in the browser, preventing the original unprotected image from being sent to a server. This protection is unobtrusive and introduces structures that degradation can be manifested through significant manipulation of the media; including cropping, resizing, recompressing, or AI-enhancements. We implemented four different methods for browser-level upload interception, utilizing dynamic input detection through a MutationObserver. We confirmed no visible perturbation and degradation was manifested through user interaction with the Extension for tamper detection verification.

Keywords: *Adversarial perturbation, browser extension, image privacy, Manifest V3, image protection, tamper detection.*

DOI: https://doi.org/10.65180/ijemri.2026.2.3.04

Introduction

With the introduction of AI-based image recognition on commercial platforms, the meaning of uploading a photo has changed. Photos sent to chatbots, social networks and email services are typically scanned by convolutional neural networks to identify faces, recognize objects and collect information on the user and all of this can happen without the user knowing the real reason behind it. Regulatory frameworks such as GDPR provide consent requirements but no technical guarantee against analysis once an image reaches a server.

Adversarial perturbation has recently emerged as more effective than traditional means. Szegedy and others say that making minimal adjustments to an image's pixels can easily trick sophisticated networks into making the wrong classification. Goodfellow built off the work of Szegedy by introducing the Fast Gradient Sign Method

(FGSM). Fawkes and LowKey are programs that disrupt facial recognition by employing the kind of perturbation that is born from the previous two studies. Unfortunately, they require access to the surrogate model, offline work, and a great deal of labor to process each image individually. None of these work through the browser in the usual upload process.

This work fills that gap with a fully client-side Chrome Extension that automatically intercepts and perturbs images at upload, applying a model-agnostic five-layer pipeline with zero server dependency and sub-200 ms processing time per image.

RELATED WORK.

Policy and Opt-Out Mechanisms

Opt-out registries maintained by face recognition providers allow people to request the removal of their images; however, users must actively upload or submit their photos to complete this process. This process inherently exposes the very image the user wishes to protect, and coverage is restricted to a single vendor's database. Regulatory instruments such as the European GDPR and the Illinois Biometric Information Privacy Act impose consent obligations but provide no signal-processing guarantee that AI analysis will not occur before an image reaches a compliant server. According to Chang et al. [11], compliance with opt-out requests varies between operators; furthermore, images that have been scraped and incorporated into training sets are seldom eliminated. Consequently, these limitations illustrate that policy cannot serve as a technical defence.

Adversarial Examples and FGSM

Early research by Szegedy et al. [1] demonstrated that very small changes to image pixels, often invisible to the human eye, can cause deep neural networks to make incorrect predictions with high confidence. Building on this research, Goodfellow et al. [2] introduced the Fast Gradient Sign Method (FGSM), a technique that creates adversarial images through slight but deliberate changes to the input image. Using the loss gradient, alterations can be made to manipulate a model to produce false outputs. Kurakin et al. [8] generalized this to a more iterative and realistic scenario, showing that adversarial perturbations can withstand multiple print and photocopy cycles. Later, Carlini and Wagner [9] illustrated that a stronger gradient-based attack could break the majority of the adversarial defences, indicating the necessity for more structural strategies over model-based strategies. These results intensely focused the theoretical perspective on pixel perturbation as a defence mechanism to privacy; however, they did not focus on the application of such perturbations in a real-time, model-agnostic, client-side manner where no access to the model gradients is provided.

Identity and Face Privacy Tools

Fawkes [3] employs a feature-space poisoning technique to facial images by creating perturbations that will maximally shift an embedded face away from its identity cluster in a face recognition system’s latent space. When tested against commercial face recognition systems, it recorded above 95% protection success rate. LowKey [4] builds upon Fawkes by implementing a more finely constrained optimization that achieves about the same protection success rate with less perturbation visibility. Alternative methods such as Yang et al. [12] used adversarial identity masks for encrypted-

style protection, synthesizing perturbed facial images using generative adversarial networks. The common drawback for all these tools is model specificity—every tool is optimized against a particular model and protection will diminish as the target model is retrained or changed. Also, all these tools fail to incorporate themselves into the browser upload pipeline. Subsequently, these tools require the user to upload images to a separate desktop application for image processing, creating an impediment for passive and everyday users.

AI Image Editing and Generation Defences

Unlike other systems, PhotoGuard [5] does not focus on recognition, but rather on AI aided image editing. Perturbations applied to protected images result in diffusion-based inpainting models producing incoherent and unsafe outputs. Glaze [13] was developed to defend against AI systems that mimic style by shifting style embeddings in the latent diffusion space. This demonstrates that adversarial perturbations can be generalized beyond classification and can be used to defend against a wider range of AI operations.

Frequency-Domain and Watermarking Approaches

The embedding of imperceptible signals into the frequency components of images for the purpose of authentication, ownership, and provenance tracking has formed the basis of a large body of research in digital watermarking. Watermarks embedded into the DCT domain at quantisation thresholds are fairly resilient to moderate compression, but unexpectedly deteriorate with geometric transformations. Design goals of spread-spectrum embedding watermarking systems are the survival of the watermark through processing as a latent and provable watermark. The system proposed in this paper inverts that design goal. Instead of surviving processing in a latent and imperceptible form, the system is designed in such a way that the embedded structure is intentionally and irreversibly distorted into visible artifacts to serve as a deterrent and a provable trace of the processing.

Browser-Level Privacy Extensions

The primary focus of previous research on browser extensions has been the prevention of tracking pixels, the removal of EXIF metadata in uploaded files, and the prevention of device fingerprinting during network requests. Since browser extensions operate at the HTTP layer, they cannot intercept the pixel-level content of images prior to their transmission, because image rendering occurs at the DOM event layer. Jha et al. proposed a client-side privacy-preserving image transformation framework. However, a functional browser extension was never developed, nor was the technology tested against real workflows on live platforms. The authors of this paper believe that this system is the first fully functional browser extension for real-time adversarial pixel perturbation, and that it operates as a Manifest V3 Chrome Extension. It can be utilized on all primary image upload pathways (file input, XMLHttpRequest, Fetch API, and a variety of other APIs).

PROPOSED SYSTEM

The extension consists of four main sections. For every page load, the content script (content.js) executes and updates file inputs with a capture-phase listener. Additionally, it modifies XMLHttpRequest.send() and window.fetch() to filter out any FormData instances that contain

image files. Single-page application frameworks add components to a web page without requiring a full refresh. To intercept these types of components, the script uses a MutationObserver. Intercepted images are sent to the adversarial engine. The engine (adversarial.js) draws the image to an off-screen Canvas, retrieves the raw image pixel buffer, applies five layers of perturbation and a marker embedding step and encodes the buffer back to JPEG format with a quality setting of 0.96. Then, it creates a new File object and replaces the original one, all before the page's own scripts get an opportunity to handle it. The background service worker (background.js) updates and syncs the state of the extension's popup and the content with the scripts. The popup (popup.html / popup.js) has an enable/disable switch, a protected-image counter, sliders for intensity and sensitivity, and a tamper verification panel.

Figure 1 shows a system architecture overview.

The upload interception layer captures images uploaded through various browsers. It observes file input, network requests, and dynamically created elements for upload. The captured images are sent to the adversarial protection engine. Here, the images undergo a multi-layered perturbation process. This is followed by inserting protection markers. The image is then re-encoded for the upload preparation process. The protected image is sent to the upload mechanism, where it replaces the original media. The upload process is completed without the need for changes to the target platform. Various supporting components (background service worker, pop-up user interface) provide tamper protection verification, protection controls, and manage extension settings, ensuring that the system operates seamlessly within the browser.

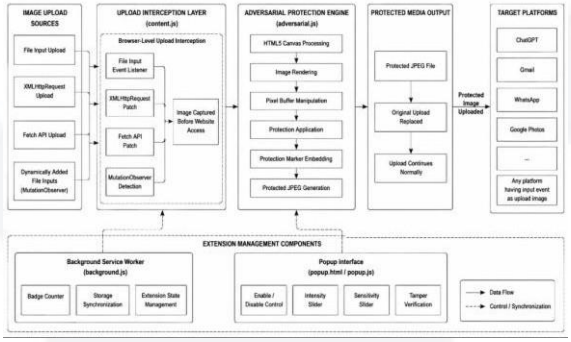


Fig. 1. Overall system architecture of the proposed image extension.

IV. METHODOLOGY

I. Layer 1: FGSM-Inspired Perturbation

Pixel brightness serves as a proxy for CNN feature magnitude. Pixels above brightness 128 receive a positive-biased perturbation; darker pixels receive a negative-biased perturbation, bounded to ±20 per channel at maximum intensity. This disrupts the low-level edge and gradient feature maps that early convolutional layers rely upon, without relying on actual backpropagation.

II. Layer 2: Cascading Pixel Dependency Chains

Each pixel's perturbation delta is a weighted linear combination of its left, top, and diagonal neighbours, modulated by a sinusoidal phase term based on position within the local 8×8 block. This encodes a spatial dependency chain: any crop or resize shifts the coordinate frame, causing cascading incorrect delta computations in all downstream pixels. JPEG recompression

compounds the effect as quantisation moves boundary values, doubling visible corruption.

III. Layer 3: Pseudo Frequency-Domain Noise

Structured cosine-based perturbation patterns inspired by frequency-domain behaviour. The amplitude sits just above the JPEG quantisation threshold — surviving the first compression but causing doubled quantisation errors on recompression, producing the characteristic 8×8 blocking pattern. AI upscalers amplify these high-frequency perturbations into ringing and colour fringing.

IV. Layer 4: Fragile Diagonal Structure Bands

Thin diagonal bands of alternating +delta/–delta pixel adjustments cancel visually due to spatial averaging by the human visual system. Any geometric transformation — crop, rotation, warp — breaks the cancellation pairs, revealing the underlying push-pull pattern as visible diagonal stripe artifacts.

V. Layer 5: Texture-Aware Embedding

A Sobel gradient approximation identifies high-texture regions. These receive stronger perturbation for three reasons: CNN feature extraction concentrates on texture and edge regions [6]; the human visual system tolerates noise in high-frequency areas better; and JPEG preserves edge regions more faithfully through multiple encode-decode cycles, ensuring perturbation longevity.

VI. Marker Embedding and Verification

After all five layers, a pseudo-random seed is encoded into the least significant bits of a small pixel region. The popup verification module cross-checks these bits against expected values. Manipulation corrupts the marker region through the cascade mechanism, producing a detectable integrity-warning status that may be classified as SUSPICIOUS or TAMPERED together with a visual corruption preview.

The sequence of operations performed by the proposed protection engine is illustrated in Fig. 2. When an image is input through the browser, the image is resized to a standard dimension. To gain direct contact to the pixel buffer, the image is processed through the HTML5 Canvas. The image then flows through five different perturbation layers. Each of these layers has a unique protection mechanism that defends against attack from AI based image analysis and perturbation. After perturbation, a protection marker is added for integrity verification. The altered image is then compressed and encoded as a JPEG and returns as a protected media object. The protected media object replaces the input image in the upload process. This system not only protects the input image from AI based image perturbation systems, but also verifies integrity against any tampering that may occur.

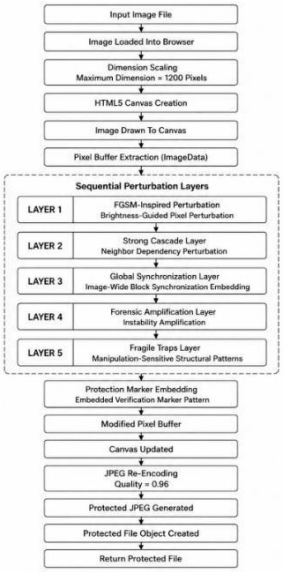


Fig. 2. Sequential perturbation layers in AdversarialProtector.protect().

V. SYSTEM MODULES

I. Adversarial Engine (adversarial.js)

Exports AdversarialProtector with protect() and verifyTamper() methods. protect() applies all five layers and returns a JPEG File. verifyTamper() reads embedded protection markers, evaluates marker integrity, and returns a tamper assessment status together with a detail string and optional corruption preview Blob.

II. Content Script (content.js)

Implements patchFileInput() for file input interception, patchXHR() and patchFetch() as immediately-invoked wrappers for network-level interception, and a MutationObserver for SPA compatibility. A guard flag on the input element prevents recursive processing when the change event is redispatched.

III. Background Worker (background.js)

Initialises defaults on install, merges new keys on update, maintains the badge counter, and handles UPDATE_BADGE, GET_STATUS, and RESET_COUNT messages. A storage change listener synchronises badge appearance with the enabled state in real time.

IV. Popup (popup.html / popup.js)

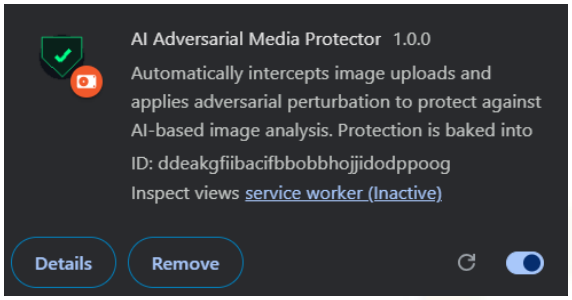
Reads saved settings from chrome.storage.local on open. Toggle and slider changes write to storage, which content scripts pick up via onChanged listeners without requiring page reload. The tamper verification drop zone runs verifyTamper() and renders results in a colour-coded panel.

VI. RESULTS AND DISCUSSION

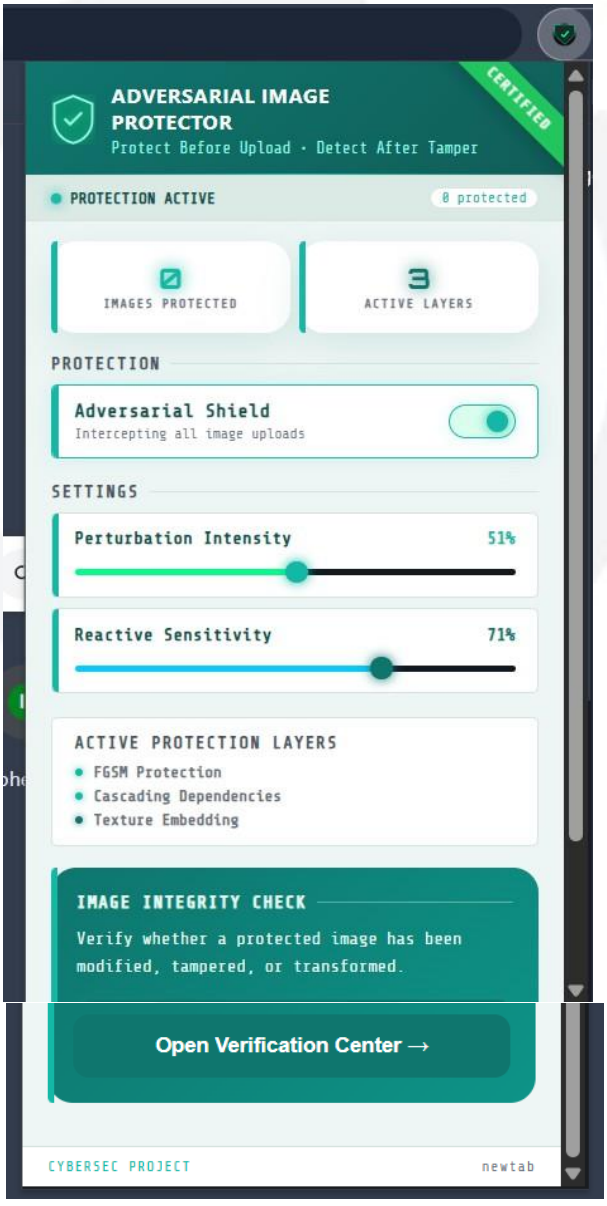
The proposed browser extension was tested to see if any image protection, verification, and tampering detection measures were available. Testing included protected images used in normal conditions, purposely manipulated images, and images subjected to various image transformations. The results show that the proposed solution successfully preserves image integrity, identifies image alterations, and offers useful image verification forensics.

The main user interface of the intended browser extension is displayed in Fig. 3. The extension is first installed and triggered in the Chrome environment as shown by Fig. 3(a), and automatically protects uploaded images. After triggering the extension, users are provided with a dashboard depicted in Fig. 3(b) that allows users to control protection settings, and manage the level of

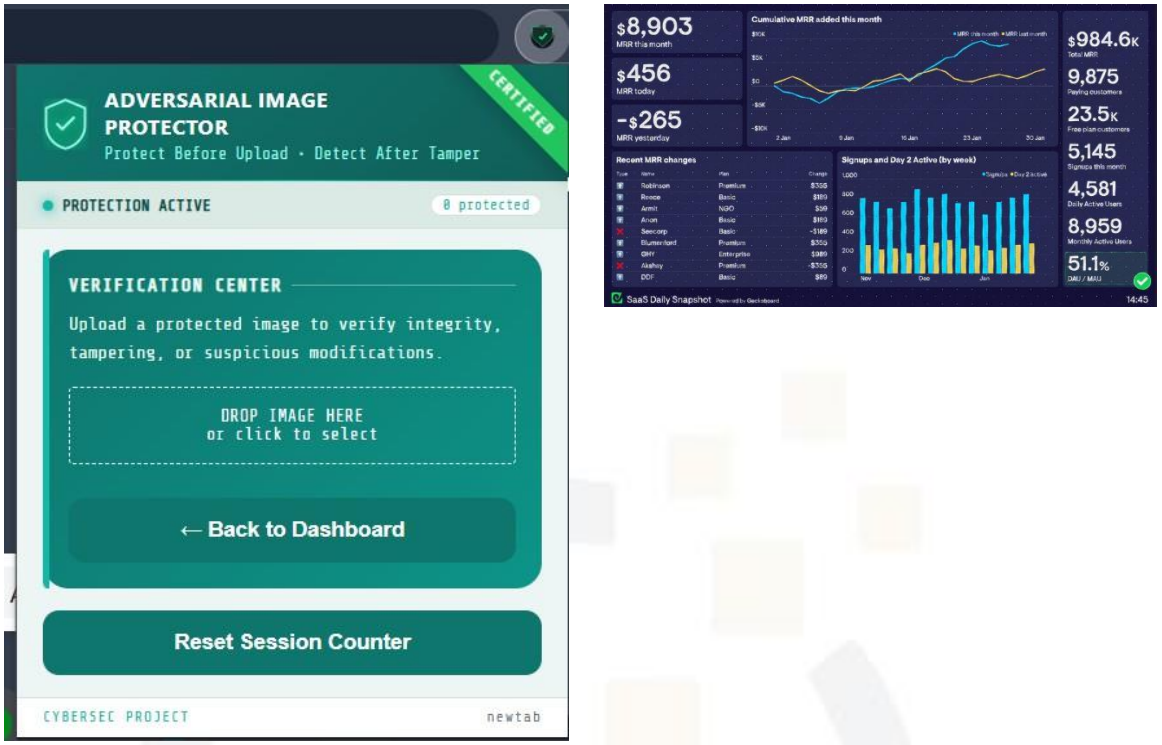
perturbation and verification sensitivity as well as the number of protection layers currently in effect. The verification center in Fig. 3(c) provides a separate setting for users to evaluate the integrity of their protected images by submitting them for authentication and tampering analysis.



A. Extension registration and activation in Chrome.



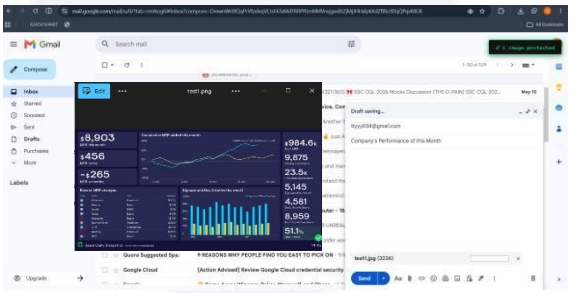
(b) Main extension dashboard showing protection controls



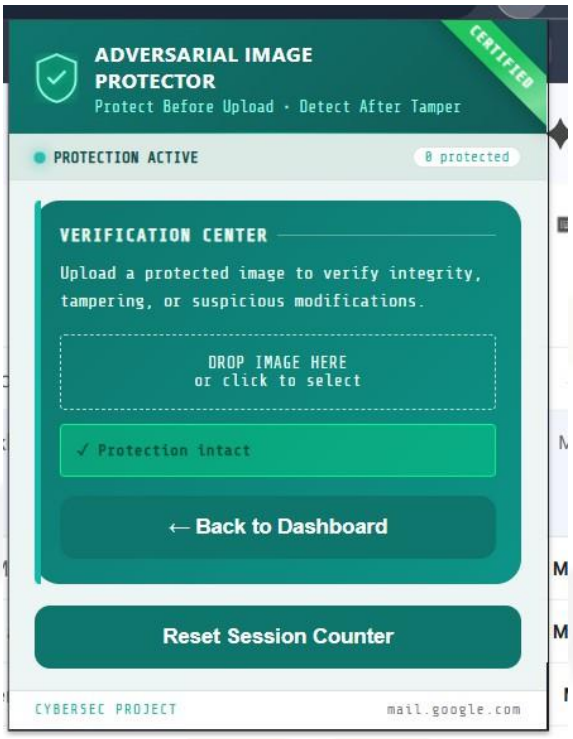
(c) Verification center for integrity and tamper verification.

Fig. 3. Adversarial image integrity and tamper check browser extension interface.

The image protection workflow illustrates in Fig. 4. As part of the protection mechanism during the image upload stage, the extension captures images and employs the suggested multi-layer protection mechanism prior to the image transmission process (see Fig. 4 (a)). The resultant protected image (see Fig. 4 (b)) appears the same as the original image, but has protection structures embedded within. Following the verification process, it is confirmed that the protection structures are intact and no modifications to the original protected image have been made. Therefore, the image successfully passes the authenticity test as shown in Fig. 4 (c).



(a) Protected image upload workflow in Gmail showing real-time upload interception



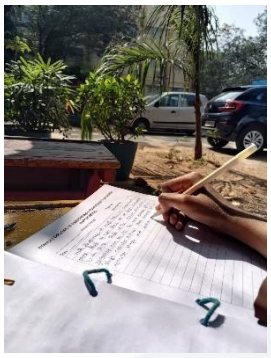
(b) Protected image after adversarial embedding and media protection processing.

(c) Verification result confirming that protection remains intact.

Fig. 4. Automatic Image Protection During Upload and Integrity Verification.

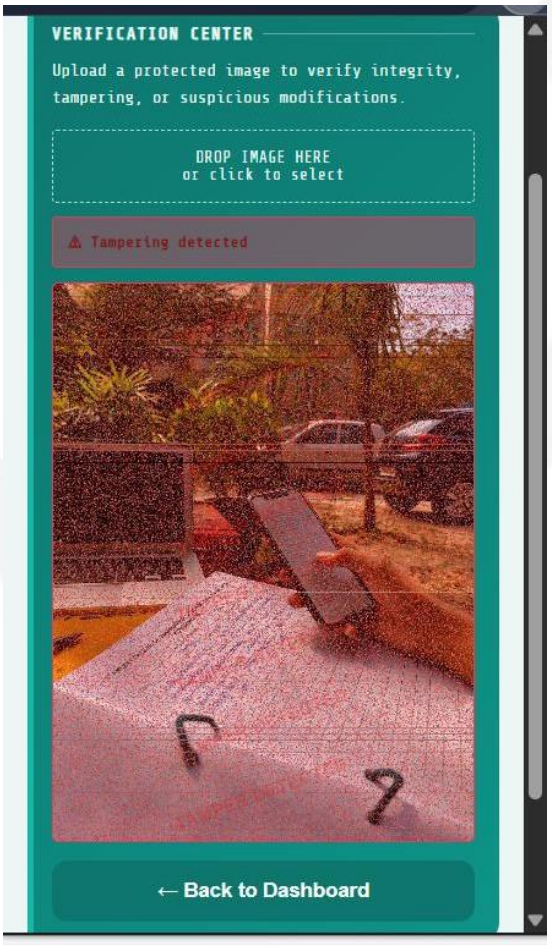
Fig. 5 shows how the new system responds to an unauthorized modification of a protected image. In Fig. 5(a), an example of the original modification scenario is illustrated. During the modification of protected media, substantial alterations to the embedded media protection structures are introduced, which ultimately leads to the failure of the media integrity. As stated in Fig. 5(c), the verification results show that the media is confirmed to be tampered with. Subsequently, the forensic corruption rendering process is activated, which provides a clear warning to the users that the protected media has been altered in an unauthorized way.

Not total tampering. This system is designed to react by explaining the verification with more detailed feedback



(a). Original image.

(b) Modified version of the image containing ...unauthorized alterations



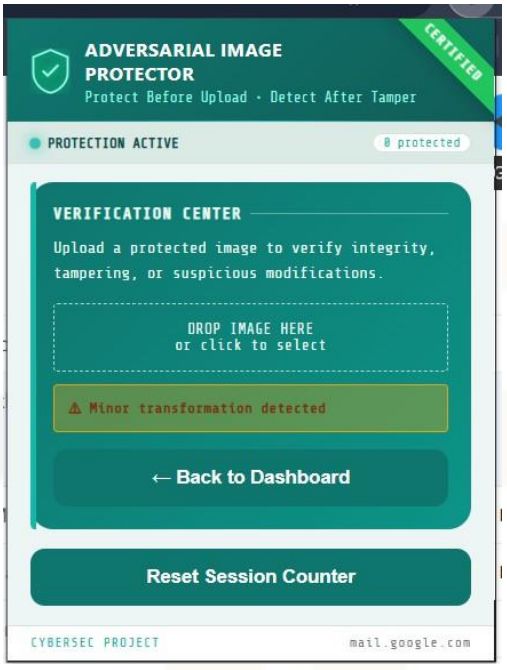
(c) Tamper verification result indicating manipulation detection and forensic corruption rendering.

Fig. 5. Tamper Detection Using Protected Media Verification.

Fig. 6 examines how the proposed system reacts to mild transformations to the original image. The image in Fig. 6(a) was cropped earning the embedded protection structures only partial destruction. Because of that, the process of

verification shown in Fig. 6(b) recorded only a minor transformation and

(a) Image subjected to cropping-based transformation.



(b) Verification result showing detection of a suspicious transformation

Fig. 6. Minor Transformation Analysis and Detection.

Verification outcomes under various image conditions are compared in Fig. 7. The left result shows the verified image in its unedited form. The image marker protection structures remain intact so this image is verified as authentic. The image in the center shows the results of a significant manipulation. Here the protection markers are disrupted and the image is classified as tampered yielding forensic corruption. The image on the right experienced some adjustments that caused only a minor disruption of the protection structures. This image is classified as suspicious. These results show that the proposed image verification framework is capable of providing a graded integrity assessment based on the level of modification.

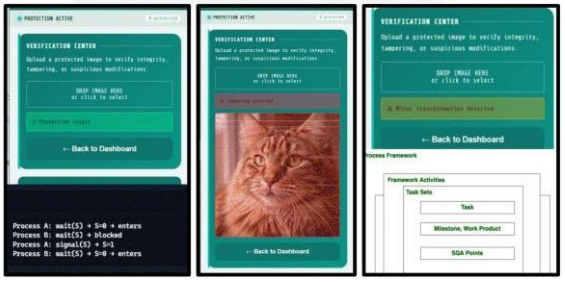


Fig. 7. Left: Image classified as protection intact. Centre: Image classified as tampered with forensic corruption indicators. Right: Image classified as suspicious due to minor transformation.

Overall, the experimented results confirm the effectiveness of the proposed browser-based

protection framework. The framework has shown the capability to protect images from being tampered with during the upload process, maintain the usability of images under standard circumstances, and identify if any integrity violations have occurred as a result of inconsistent modifications. The framework's verification mechanism provided useful forensic feedback corresponding to the severity of the detected alterations. This shows the proposed approach is practical for image authenticity protection and verification.

VII. CONCLUSION

This work introduced a practical approach to combine tamper detection with adversarial image protection in a way that is usable by web browsers. The proposed system uses three components: multiple perturbation layers to reduce the effectiveness of automated AI-based image analysis and feature extraction; tamper sensitive structures that are embedded within an image so they can be used to verify if an image has been altered after it leaves an organization's control; and uploading intercepts at the level of the browser to prevent a protected image from being uploaded. The system protects images before they are transmitted and allows an organization to perform a post-transmission integrity check via forensic analysis. The study demonstrates that both meaningful protection and authenticating a legitimate image can be accomplished in real-time using this framework without the need for special platforms, remote processing architecture or modification to current online service providers. Current and future enhancements include increasing resistance to newer forms of image transformation (e.g., AI-generated images) as well as expanding the framework to protect other types of media.

REFERENCES

1. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," arXiv preprint arXiv:1312.6199, 2013.
2. Goodfellow IJ, Shlens J, Szegedy C. “Explaining and harnessing adversarial examples.” arXiv preprint arXiv:1412.6572. 2014 Dec 20.
3. Shan S, Wenger E, Zhang J, Li H, Zheng H, Zhao BY. “Fawkes: Protecting privacy against unauthorized deep learning models.” In29th USENIX security Symposium (USENIX Security 20) 2020 (pp. 1589-1604).
4. Cherepanova V, Goldblum M, Foley H, Duan S, Dickerson J, Taylor G, Goldstein T. “Lowkey: Leveraging adversarial attacks to protect social media users from facial recognition.” arXiv preprint arXiv:2101.07922. 2021 Jan 20.
5. Radiya-Dixit E, Hong S, Carlini N, Tramer F. “Data poisoning won't save you from facial recognition.” arXiv preprint arXiv:2106.14851. 2021 Jun 28.
6. Geirhos R, Rubisch P, Michaelis C, Bethge M, Wichmann FA, Brendel W. “ImageNet-trained CNNs are biased towards texture;” increasing shape bias improves accuracy and robustness. In International Conference on Learning Representations 2018 Nov 29.

7. Hore A, Ziou D. “Image quality metrics: PSNR vs. SSIM.” In2010 20th international Conference on Pattern Recognition 2010 August 23 (pp. 2366-2369). IEEE.
8. Kurakin A, Goodfellow IJ, Bengio S. “Adversarial examples in the physical world.” In Artificial intelligence safety and security 2018 Jul 27 (pp. 99-112). Chapman and Hall/CRC.
9. Carlini N, Wagner D. “Towards evaluating the robustness of neural networks.” In2017 IEEE Symposium on Security and Privacy (SP) 2017 May 22 (pp. 39-57). Ieee.
10. Xie C, Wang J, Zhang Z, Zhou Y, Xie L, Yuille A. “Adversarial examples for semantic segmentation and object detection.” In Proceedings of the IEEE International Conference on Computer Vision 2017 (pp. 1369-1378).
11. J. Chang, V. Liberatore, and M. Liberatore, "Compliance and effectiveness of facial recognition opt-out mechanisms," in Proc. Privacy Enhancing Technologies Symposium (PETS), 2022, pp. 412–431.
12. X. Yang, Y. Dong, J. Pang, T. Zhu, and J. Zhu, "Towards face encryption by generating adversarial identity masks," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2021, pp. 3897–3906.
13. S. Zhao, N. Ruiz, and J. B. Simon, "Glaze: Protecting artists from style mimicry by text-to-image models," in Proc. 32nd USENIX Security Symposium, 2023, pp. 2187–2204.
14. I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, “Digital Watermarking and Steganography”, 2nd ed. Burlington, MA: Morgan Kaufmann, 2008.
15. S. Jha, Z. Hayashi, and W. Rivoire, "Privacy-preserving image transformation at the browser level: A conceptual framework," in Proc. IEEE Int. Workshop on Information Forensics and Security (WIFS), 2020, pp. 1–6.