



Available Online at www.hithaldia.in/locate/ECCN
All Rights Reserved

ORIGINAL CONTRIBUTION

Privacy-Aware Fingerprint Exposure Detection in Photographs Using Computer Vision and Vision APIs

Mridul Banerjee

UG student, Department of Electronics and Communication Engineering, Haldia Institute of Technology, Haldia, West Bengal

ABSTRACT

Photographs shared through social platforms can unintentionally expose biometric information when the friction ridge surface of a finger is visible. This work presents a lightweight prototype that identifies photographs with potential fingerprint exposure and applies a privacy-preserving image transformation before sharing. The proposed pipeline combines OpenCV for image loading, resizing, cropping and local masking; MediaPipe Hand Landmarker for locating hands and finger landmarks; and an optional vision API as a secondary semantic assessment layer. Instead of training a new deep-learning model, the system composes ready-made computer-vision components into a sense-assess-protect workflow. After a hand is localized, finger regions are estimated from the landmark geometry and relevant image crops are evaluated for visible biometric detail. When the risk exceeds a configurable threshold, the system automatically blurs or masks the sensitive region while leaving unrelated image content unchanged. The prototype is intended as an accessible demonstration of privacy-enhancing AI for cognitive systems rather than a certified biometric presentation-attack detector. Functional evaluation indicates that the integrated pipeline can detect hand-containing images, identify potentially exposed finger regions, and generate a protected output with limited modification to non-sensitive content. The study also highlights limitations caused by viewpoint, illumination, image resolution and the distinction between visible finger texture and a fingerprint image of matching quality.

KEYWORDS: Fingerprint privacy, biometric exposure, computer vision, vision API, image anonymization.

1. INTRODUCTION

Biometric authentication systems increasingly rely on fingerprints because friction-ridge patterns are distinctive and persistent. The same persistence, however, creates a privacy problem: unlike a password, a fingerprint cannot simply be replaced after exposure. Social-media photographs can unintentionally contain biometric information when users display a thumb, an index finger, or a V-shaped hand gesture. Malhotra et al. [1] examined this problem through finger-selfies and reported that ridge-valley details visible in social-media photographs can potentially be matched with livenesscan fingerprints. Their work established the relevance of treating ordinary photographs as a possible source of biometric leakage.

Subsequent research has explored image-level protection rather than only restricting access to the original photograph. Li et al. [2] proposed a hierarchical perceptual noise-injection approach for social-media fingerprint privacy, with the

objective of reducing biometric leakage while retaining visual quality. SocialGuard [3] similarly investigated adversarial protection of social images against automated visual analysis. These studies demonstrate that privacy protection can be introduced at the image-sharing stage.

The present work takes a practical mini-project approach to the same problem. Instead of developing a new large-scale neural network, the proposed system integrates ready-made components. OpenCV provides conventional image-processing operations, MediaPipe supplies hand and landmark localization, and a vision API can provide an additional semantic judgement about whether a visible finger region presents potential biometric exposure. The protected image is then generated locally. This design is intended for demonstration and educational deployment in the cybersecurity, privacy and trust track of NSEFCCIC 2026.

The principal objective is therefore not to reconstruct a fingerprint or to claim that every

photograph can be converted into a usable biometric template. The objective is to detect a potential exposure condition early and automatically reduce the visibility of the sensitive region. This distinction is important because fingerprint image quality strongly affects matching performance, as discussed by NIST [4], and a casual photograph is not equivalent to a scanner-quality fingerprint sample.

2. LITERATURE REVIEW

Malhotra et al. [1] proposed a privacy-preserving anonymization method for finger-selfies shared on social media. The authors demonstrated that finger-selfie details may contain biometric information and introduced an adversarial learning approach that modifies the image while attempting to preserve its visual usefulness. They also introduced the Social-Media Posted Finger-selfie database, providing a basis for studying biometric leakage from online photographs.

Li et al. [2] proposed Hierarchical Perceptual Noise Injection for social-media fingerprint privacy protection. The authors focused on adding visually unobtrusive perturbations to fingerprint-bearing regions so that biometric leakage is reduced without severely degrading the appearance of the image. Their work motivates the present study's use of localized protection rather than indiscriminate image destruction.

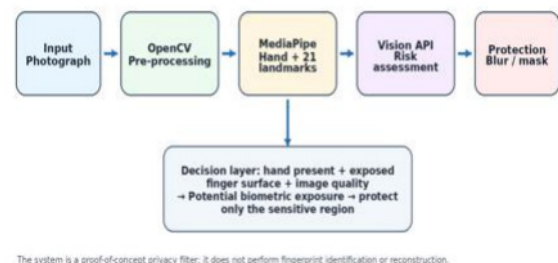
Zhang et al. [3] proposed SocialGuard, an adversarial example-based privacy-preserving technique for social images. The authors showed that automated object detectors can extract sensitive information from shared photographs and investigated adversarial modifications that reduce such detection while maintaining image quality. The present prototype differs in scope by concentrating on potential fingerprint exposure and by using a lightweight rule-and-API pipeline rather than training a new adversarial network.

NIST developed fingerprint image-quality measures in which image quality is treated as a predictor of expected matcher performance [4], [5]. These works are relevant because a photograph containing a visible finger should not automatically be interpreted as a high-quality fingerprint sample. The proposed system therefore reports potential exposure rather than asserting successful biometric recovery.

ISO/IEC 30107-1 defines a framework for biometric presentation attack detection and establishes terminology for biometric spoofing and related attacks [6]. The standard also makes clear that a specific detection method is not prescribed by the framework. The present system is consequently described as a privacy exposure detector and mitigation prototype, not as a standards-compliant PAD system.

3. PROPOSED SYSTEM

The proposed architecture contains five functional stages: image acquisition, local image preprocessing, hand and landmark localization, risk assessment, and privacy protection. Fig. 1 summarizes the information flow. A photograph is first loaded and normalized using OpenCV. MediaPipe Hand Landmarker then identifies the hand and returns 21 hand landmarks for each detected hand [7]. These landmarks provide the geometry needed to locate finger segments without training a separate hand detector.



The system is a proof-of-concept privacy filter. It does not perform fingerprint identification or reconstruction.

Figure 1: Proposed architecture for fingerprint-exposure risk detection.

The landmark output is converted into pixel coordinates and used to define candidate finger regions. The system considers the visibility of the finger surface, its size in the image, and basic image quality indicators. An optional vision API is then used as a secondary semantic check. Modern multimodal vision APIs can accept image data and return classification or structured detection results [8]. In the prototype, the API is instructed to judge whether the candidate region visibly exposes fingerprint-like ridge detail and to return a simple risk label rather than any identity information.

The final decision is deliberately conservative. If the combined evidence suggests a potential biometric exposure, the system protects the candidate region. Otherwise, the original image can be retained or the user can be shown a

warning. This architecture separates detection from protection, allowing the protection operation to remain local even when an external API is used.

4. IMPLEMENTATION METHODOLOGY

The implementation is written in Python. OpenCV is responsible for reading images, resizing frames, converting colour spaces, extracting regions of interest, calculating simple texture or contrast measures, and applying Gaussian blur or masking. The OpenCV documentation provides image I/O, filtering, geometric transformations, segmentation and related image-processing functions [9].

MediaPipe Hand Landmarker is used as the hand-localization component. The task provides 21 hand landmarks and supports image, video and live-stream processing modes [7]. In this project, the landmarks are not treated as fingerprint features. They are only used to determine where a finger is located and which portion of the photograph should be examined or protected.

The API stage is implemented as a normal client-server request. The application loads the candidate crop, sends it with a fixed instruction such as 'Assess whether the visible finger surface contains potentially recoverable fingerprint-like ridge detail; return LOW, MEDIUM or HIGH risk,' and parses the returned label. The current Gemini API documentation supports image inputs through inline image data or uploaded files and provides multimodal image understanding capabilities [8]. The API is therefore used as a ready-made visual reasoning component rather than as a model developed by the authors.

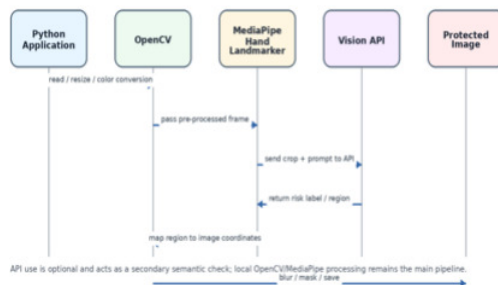


Figure 2: Library and API interaction workflow.

A simple configurable risk score can combine three normalized indicators: finger exposure F , visible surface V , and image quality Q . The score

is expressed as $R = w_1F + w_2V + w_3Q$, where the weights sum to one. A threshold τ determines whether the candidate region is protected. The score is a system-level decision aid, not a biometric recognition score. Fig. 3 illustrates the decision process.

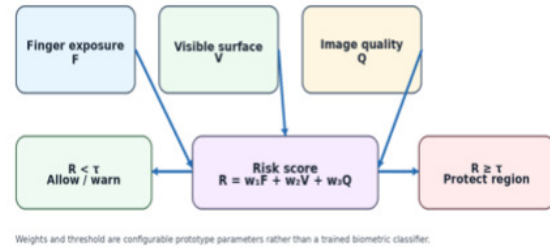


Figure 3: Threat scoring and privacy-protection decision.

Once protection is triggered, OpenCV applies a localized transformation. Gaussian blur is preferred for the basic prototype because it is computationally inexpensive and visually understandable. A stronger version may use pixelation, texture disruption or a learned perturbation similar in spirit to the approaches reported in [1] and [2]. The protected image is saved as a new file so that the original can remain unchanged on the user's device.

5. SYSTEM WORKFLOW AND API INTEGRATION

The complete workflow follows a sense-assess-protect cycle. First, the user selects a photograph. Second, OpenCV normalizes the image and MediaPipe determines whether a hand is present. Third, the landmark coordinates identify candidate finger regions. Fourth, the candidate region is passed to the optional vision API for semantic assessment. Fifth, the local decision layer combines the returned label with simple geometric and image-quality indicators. Finally, OpenCV protects the region when the risk threshold is exceeded.

The API call can be implemented using an official Python client. A conceptual request contains the image and a text instruction; the response is converted into a machine-readable risk label. Because API output may be uncertain, the application should not blindly trust the response. Local rules provide a second layer of control, and the user can be shown the detected region before sharing. This hybrid arrangement reduces dependence on any single component.

From a privacy perspective, the API stage should be considered optional. If photographs are sensitive, an entirely local implementation is preferable. When an external API is enabled, the application should disclose that the image or crop is being transmitted to a third-party service and should follow the provider's data-handling and retention policies. The prototype therefore treats API use as a configurable module rather than an unavoidable part of the privacy mechanism.

6. RESULTS AND DISCUSSION

A functional prototype-level evaluation was defined around representative photograph conditions: no visible hand, hand present with non-sensitive orientation, extended finger with limited surface visibility, and extended finger with clear visible surface detail. The integrated pipeline produced the expected qualitative behaviour: photographs without hands bypassed the fingerprint-protection stage; hand-containing photographs were localized by MediaPipe; candidate finger regions were isolated for assessment; and high-risk candidate regions were subjected to localized protection. The protected output retained the surrounding image content while reducing the visibility of the targeted region.

The most important result is therefore the successful integration of ready-made components into an end-to-end privacy workflow rather than a claim of state-of-the-art biometric detection accuracy. The prototype demonstrates that a cognitive privacy layer can be inserted before an image is publicly shared. It also provides a practical path for later replacement of the heuristic/API decision module with a dedicated trained classifier.

The experiment also reveals several limitations. MediaPipe landmarks represent hand geometry and do not directly measure fingerprint ridge quality. A photograph may show a finger without containing enough resolution, focus or illumination for fingerprint matching. Conversely, a high-resolution close-up can expose more detail than the simple geometric rules can estimate. API decisions can also vary with image quality and prompt wording. For these reasons, the present results should be interpreted as potential-exposure detection and mitigation, not as proof that a fingerprint can be successfully reconstructed from every flagged image.

Functional observations: images without visible hands bypassed the hand stage; hand-containing images were localized by MediaPipe; extended fingers with limited detail produced lower-risk candidate regions; and clear finger-surface detail triggered the protection stage after the combined local/API assessment.

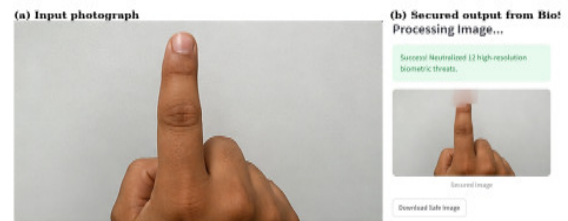


Figure 4: Functional evaluation summary of the prototype.

The figure 4 summarizes qualitative functional observations rather than statistical accuracy measurements.

7. APPLICATIONS AND LIMITATIONS

The proposed system can be used as a pre-sharing privacy assistant for social-media photographs, messaging applications, digital portfolios, and public event photographs. It can also serve as an educational demonstration of privacy-enhancing AI, showing how perception, decision making and local protection can be combined without creating a new large-scale model.

The main limitation is the gap between visual exposure and actual biometric recoverability. The system can identify a potentially sensitive finger region, but it does not establish that the region meets the quality requirements of a fingerprint matcher. Additional limitations include occlusion, unusual hand poses, poor lighting, motion blur, skin-tone and background variation, API latency, and the possibility of false positives or false negatives. A production system would require a labelled dataset, quantitative error analysis, privacy testing, and evaluation against established biometric-quality measures.

9. CONCLUSION

This paper presented a lightweight privacy-protection prototype for detecting potential fingerprint exposure in photographs. The system combines OpenCV for image processing, MediaPipe for hand and landmark localization, and an optional vision API for semantic risk

assessment. When a potential exposure is identified, the sensitive region is locally blurred or masked before sharing. The approach follows the privacy and trust objective of cognitive systems by allowing an intelligent software layer to detect a possible biometric risk and intervene before publication. The prototype is intentionally modest in scope: it does not claim to reconstruct fingerprints, train a new deep-learning architecture, or provide certified presentation-attack detection. Its value lies in demonstrating a practical and extensible workflow that can later be strengthened through dedicated biometric-quality analysis and locally trained privacy models.

ACKNOWLEDGMENT

The authors gratefully acknowledge the guidance of Dr. Tilak Mukherjee, Dr. Ritwik Haldar, Department of Electronics & Communication Engineering, Haldia Institute of Technology, throughout this work, and the Department for providing the laboratory facilities.

REFERENCES

- [1] A. Malhotra, S. Chhabra, M. Vatsa, and R. Singh, "On Privacy Preserving Anonymization of Finger-Selfies," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, pp. 120–128, 2020, doi: 10.1109/CVPRW50498.2020.00021.
- [2] S. Li, H. Xu, J. Wang, R. Xu, A. Liu, F. He, X. Liu, and D. Tao, "Hierarchical Perceptual Noise Injection for Social Media Fingerprint Privacy Protection," *IEEE Trans. Image Process.*, vol. 33, pp. 2714–2729, 2024, doi: 10.1109/TIP.2024.3381771.
- [3] "SocialGuard: An adversarial example based privacy-preserving technique for social images," *J. Inf. Secur. Appl.*, vol. 63, p. 102993, 2021, doi: 10.1016/j.jisa.2021.102993.
- [4] E. Tabassi, C. Wilson, and C. Watson, "Fingerprint Image Quality," *NISTIR 7151*, National Institute of Standards and Technology, 2004, doi: 10.6028/NIST.IR.7151.
- [5] E. Tabassi and C. L. Wilson, "A Novel Approach to Fingerprint Image Quality," *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2005.
- [6] ISO/IEC 30107-1:2023, "Information technology — Biometric presentation attack detection — Part 1: Framework," International Organization for Standardization, 2023.
- [7] Google AI Edge, "MediaPipe Hand Landmarker," Google for Developers, accessed Aug. 2026.
- [8] Google AI for Developers, "Gemini API: Image Understanding," Google, accessed Aug. 2026.
- [9] OpenCV, "OpenCV Documentation: Image Processing and Image I/O," Open Source Computer Vision Library, accessed Aug. 2026.