



A ROBUST FRAMEWORK FOR COPYRIGHT PROTECTION AND OWNERSHIP VERIFICATION OF DIGITAL IMAGES

Pavithra N¹, Anushree P M², Kavana K³, Megha Manjappa Mogali⁴, V Tejaswini⁵

Assistant Professor, Department of Computer Science & Engineering, S.J.M Institute of Technology, Chitradurga,
Karnataka, India¹

B.E. Student, Department of Computer Science & Engineering, S.J.M Institute of Technology, Chitradurga, Karnataka,
India^{2,3,4,5}

Abstract: The widespread sharing of digital images through websites and social media has made unauthorized copying, redistribution, and manipulation easier. These issues create difficulties for photographers, designers, content creators, and organizations seeking to establish ownership of their work. This paper presents a robust framework for copyright protection and ownership verification of digital images using digital watermarking and cryptographic hashing. The proposed system is designed to register an image with creator metadata, embed copyright information as a watermark, generate a SHA-256 hash, and store the related ownership record for later verification. During verification, the submitted image is checked against stored records; a hash mismatch indicates that the file differs from the registered version, while watermark extraction can provide an additional ownership check when supported by the implemented watermarking method. The framework is planned around Python and image- processing libraries, with a database for ownership metadata. The design aims to combine image-level copyright marking with cryptographic integrity checking. Quantitative performance and robustness must be established through implementation and testing; this paper therefore does not claim unmeasured accuracy or performance results.

Keywords: Digital image copyright, digital watermarking, SHA-256, cryptographic hashing, ownership verification, image integrity, Python.

I. INTRODUCTION

Digital images are widely used in education, journalism, advertising, entertainment, research, and social media. Digital platforms allow images to be copied and distributed quickly, often without the creator's permission or attribution. Because identical copies can be produced without visible quality loss, it can be difficult to determine the original owner or establish whether a submitted file has been altered.

Copyright protection for digital images can be supported by technical mechanisms that associate ownership information with the image and provide away to verify its integrity. Digital watermarking embeds information into the image, while cryptographic hashing generates a digest that can be compared with a previously stored digest. These methods address related but different needs: watermarking can carry ownership information within media, where as cryptographic hash can identify whether a file is exactly the same as a registered file.

This project proposes a Python-based framework combining watermarking, SHA-256 hashing, metadata storage, and an ownership-verification module. The intended workflow enables a creator to register an image and later submit an image for verification against the stored ownership record.

II. RELATED WORK

Cox, Miller, and Bloom describe foundational watermarking concepts and methods for embedding information in digital media [1]. Barni, Bartolini, and Piva discuss wavelet-based watermarking and perceptual masking, illustrating the need to balance watermark strength with visual quality [2]. Hsu and Wu examine hidden digital watermarks in images, which is relevant to invisible ownership marks [3]. Johnson and Jajodia discuss steganography and the concealment of information [4]. Fridrich surveys digital image forensics and techniques that can help identify image manipulation [5]. These works provide background for combining embedded ownership information with integrity



checks. The project report also lists newer work on hybrid watermarking, block chain records, and learning-based watermarking, but some bibliographic details are incomplete or placeholder-like; those entries are not treated as verified citations in this paper. The design gap addressed here is the need for a straightforward, modular workflow that combines watermarking, exact-file integrity checking, persistent ownership metadata, and a user-facing verification result.

III. PROBLEM STATEMENT AND OBJECTIVES

Digital creators face risks from unauthorized copying, distribution, and modification of images. A practical system is needed to register ownership metadata, attach watermark-based evidence, check file integrity, and retrieve verification results efficiently. The system provides technical support and does not replace legal copyright registration or determine legal ownership by itself.

OBJECTIVES:

- Design a Python-based system for image copyright protection using watermarking and cryptographic hashing.
- Provide a registration process that stores creator details, image metadata, and the generated hash.
- Implement ownership verification by comparing submitted image data with stored records.

IV. PROPOSED SYSTEM ARCHITECTURE

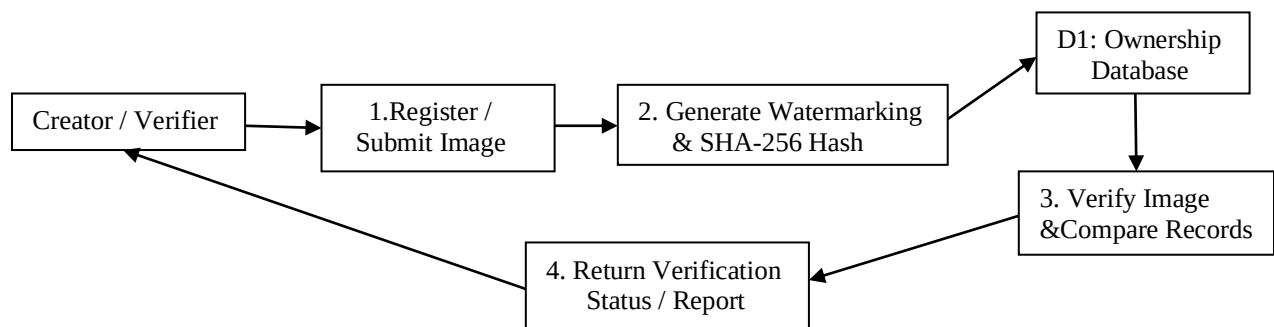


Fig. 2. Proposed data flow for image registration, hash storage, and ownership verification.

A. User Interface Module

The interface allows a creator to upload an image, enter ownership metadata, request watermark embedding, and submit an image for verification. The project report identifies Flask or Django for a web interface, or Tkinter for a desktop interface, as possible options. The final implementation should select one technology and document its version.

B. Watermarking and Hashing Module

The watermarking component embeds an ownership identifier or compact ownership reference using a specified algorithm. OpenCV or Pillow may support image processing. The hashing component computes a SHA-256 digest of the registered file using Python's hashlib library. The digest is stored with the registration record and compared with the digest of a submitted file.

C. Ownership Database

The database stores a content identifier, creator metadata, registration timestamp, watermark reference, SHA-256 digest, and verification history. The report proposes MySQL or PostgreSQL. Access control, protected credentials, secure backups, and encryption for sensitive metadata should be included.

D. Verification and Tamper Analysis

The verification component retrieves the ownership record, recomputes the submitted file's SHA-256 digest, and compares it with the stored value. It also checks the watermark according to the selected embedding and extraction method. A hash mismatch means the file bytes differ; it does not by itself prove malicious tampering. Any machine learning classifier is an additional module that must be evaluated separately. E. Reporting Module The reporting component presents results such as exact-file match, hash mismatch, watermark detected, watermark not detected, or

insufficient evidence. It should record the request time and the ownership record examined, and avoid claiming legal ownership solely from a matching hash or watermark.



Proposed Workflow: 1) Upload image and metadata → 2) Embed ownership watermark → 3) Calculate SHA-256 → 4) Store ownership record → 5) Submit image for verification → 6) Recalculate SHA-256 → 7) Check watermark and stored record → 8) Display result or alert.

V. METHODOLOGY

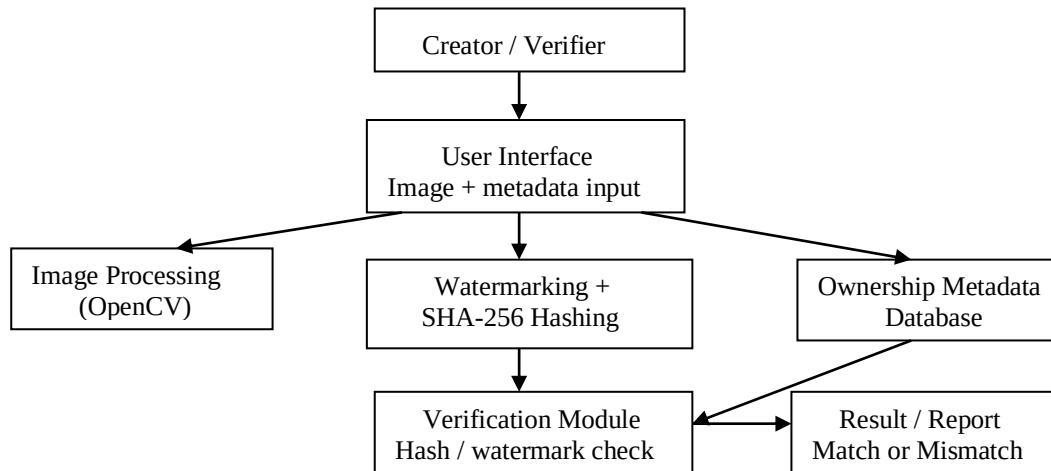


Fig.1. Proposed system architecture for digital image copyright protection and Ownership verification.

A. Image Registration

The user selects an image and provides ownership metadata. The system validates file type and size, assigns a unique registration identifier, and prepares a copy for watermark processing. The original file should be retained according to a documented storage policy so later checks can be reproduced.

B. Watermark Embedding

The watermark payload should identify the registered record or encode a compact ownership reference. The implementation must specify the embedding method, payload format, embedding strength, extraction procedure, and supported image formats. Invisible watermarking involves a trade-off between imperceptibility and robustness.

C. Hash Generation and Storage

After the registered file version is finalized, the system calculates its SHA-256 digest and stores it with the ownership metadata. The system must state whether it hashes the original image or the watermarked output. To verify an exact registered file, hashing must be performed on the same byte sequence that was registered.

D. Ownership Verification

For a verification request, the system locates the ownership record and computes the digest of the submitted file. A matching digest supports an exact-file match. A mismatch indicates a byte-level difference and triggers further checks, including watermark extraction where supported. Resizing, compression, or re-encoding usually changes the hash even if the image appears visually similar; robust ownership checks therefore depend on the watermark method rather than the hash alone.

E. Machine-Learning Extension

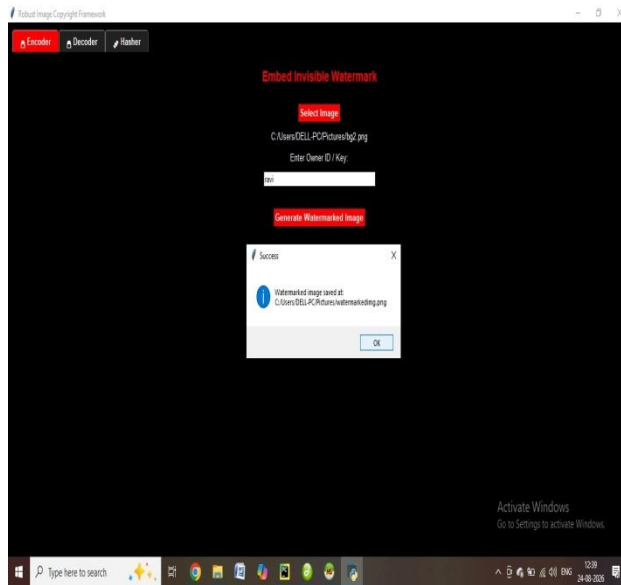
The project report proposes supervised methods such as Support Vector Machines, k-Nearest Neighbours, or Logistic Regression to classify content as original or tampered. This is an implementation option, not a demonstrated result in the available report. If used, the dataset should contain labelled original and transformed/manipulated images, with training and test sets separated by source image. Precision, recall, F1-score, confusion matrix, and false positive rate should be reported.

VI. RESULTS AND DISCUSSION

The project design defines a workflow for registering images, storing ownership-related metadata and SHA-256



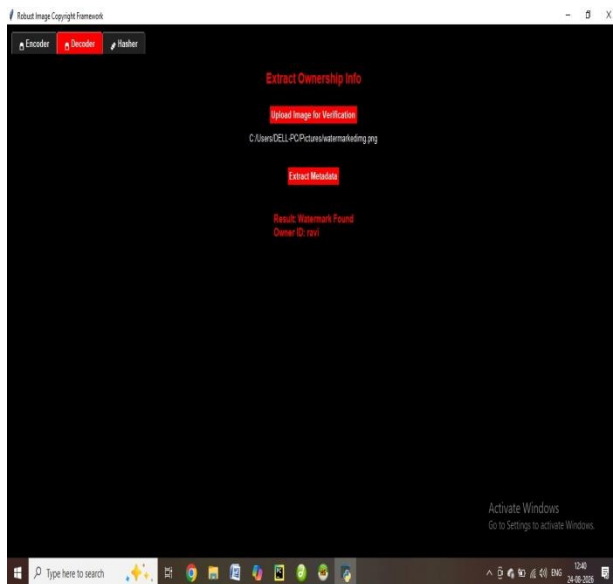
digests, and checking submitted images against registered records. For an unchanged file, the expected behaviour is that there computed digest matches the stored digest. If the file is modified, the digest is expected to differ, prompting the system to report a mismatch. If watermark embedding and extraction are implemented successfully, the watermark can provide an additional ownership signal.



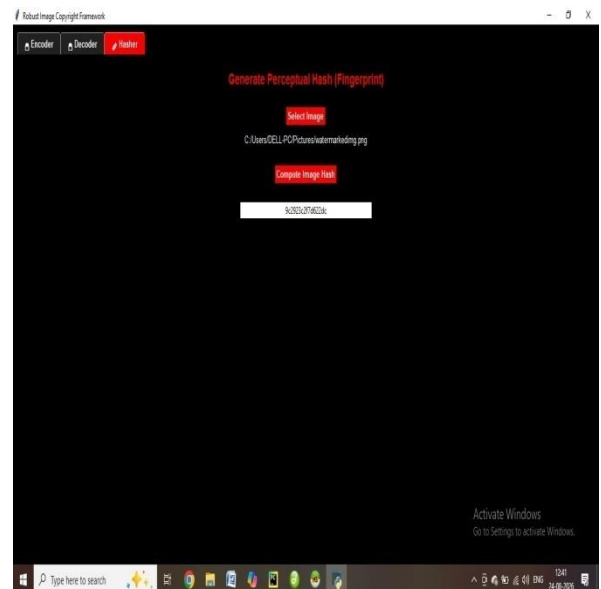
(a)



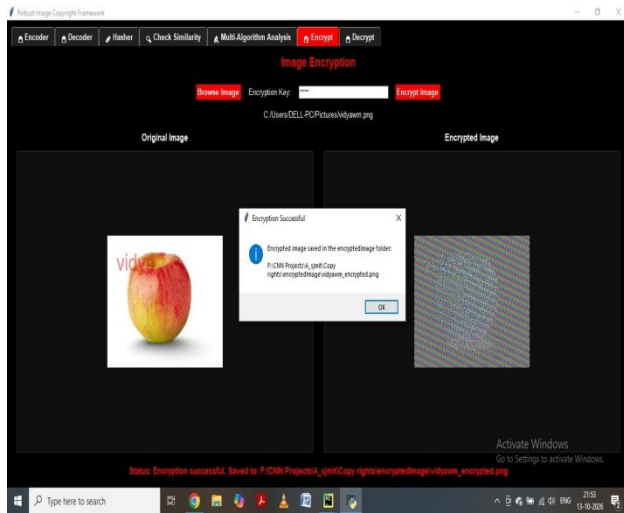
(b)



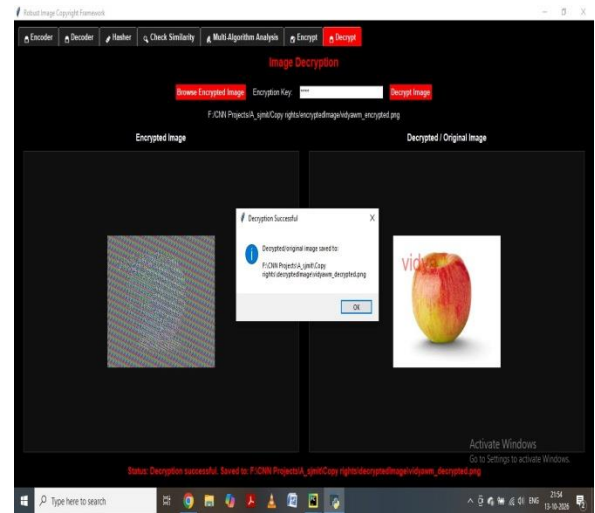
(c)



(d)



(e)



(f)

Fig. 3. Proposed Robust Image Copyright Protection and Ownership Verification Framework: (a) Invisible Watermark Embedding Using the Encoder Module (b) Watermarked Image Showing Embedded Owner Information (c) Ownership Verification and Watermark Extraction Using the Decoder Module (d) Perceptual Hash Generation for Digital Image Fingerprinting (e) Image Encryption Using the Proposed Framework (f) Image Decryption and Original Image Recovery.

VII. CONCLUSIONS

This paper presents a framework for digital image copyright protection and ownership verification that combines watermarking, SHA-256 hashing, metadata storage, and a verification workflow. Watermarking is intended to associate ownership information with an image, while cryptographic hashing provides an exact-file integrity check against a registered digest. The framework is designed using Python and image-processing and cryptographic libraries, with a database for ownership records. Its effectiveness will depend on implementation quality, database security, and empirical testing. Future work may evaluate different watermarking methods, add stronger authentication and audit logging, and investigate machine-learning or block chain-based extensions, while reporting results only after experimental validation.

ACKNOWLEDGMENT

The authors express their sincere gratitude to the management of S.J.M. Vidyapeetha, the Principal, the Head of the Department of Computer Science and Engineering, the project coordinator, and the faculty members of S.J.M. Institute of Technology, Chitradurga, for their guidance and support. The authors especially thank their project guide, Prof. Pavithra N, for valuable suggestions and encouragement throughout the project work.

REFERENCES

- [1] M. Begum and M. S. Uddin, "Digital Image Watermarking Techniques: A Review, " 2020. Complete publication details should be verified from the original source.
- [2] P. A. C. And M. T. Somashekara, "Watermarking for Tamper Detection, Copyright Protection, and Rightful Ownership of a Digital Image," 2022. Complete publication details should be verified from the original source.
- [3] M. Mrak and J. Flusser, "Digital Image Integrity– A Survey of Protection and Verification Techniques," 2017. Complete publication details should be verified from the original source.
- [4] "Digital Image Watermarking Using Deep Learning: A Survey," 2024. Author a Details should be verified from the original source.
- [5] PythonSoftwareFoundation, "hashlib—Securehashesandmessagedigests," Pythondocumentation.
- [6] Open CV, "Open CV Documentation," official documentation.