Self Embedding Watermarking System

Software Requirement Specifications

Project Advisor:

Dr. Asif Mahmood Gilani

Group Members:

Arbab Hamd Rizwan	18L-0756
Usama Aslam	18L-0972
Aashar Naseem	18L-1131
M. Hunzlah Malik	18L-1139

National University of Computer and Emerging Sciences

Department of Computer Science

Lahore, Pakistan

Table of Contents

List of Tables

List of Figures

1. Introduction

Editing software has come a long way in the last decade. As a result, the general public now has access to complex editing tools, making it easier to manipulate digital assets like photographs and movies. This casts considerable question on the credibility of any media offered. Image tampering is primarily concerned with changing the image as a whole. Video, on the other hand, is made up of multiple frames that can be regarded as individual images.

Therefore, video tampering has been categorized into two types: [1]

1. Temporal tampering refers to interframe editing manipulation. This type of tampering includes adding, removing, or changing the sequence of frames in a video.
2. Spatial tampering includes manipulating objects within a frame and is referred to as intraframe manipulation.

Watermarking is a technique that can be used to detect tampering. Our project will be focused on detecting and restoring these altered images and videos. Watermarking images entails breaking them down into smaller chunks and inserting data into the LSB, which can then be used to detect alteration and restore the original image. Numerous approaches are investigated in research publications, varying depending on the number of blocks the image is divided into and the watermarking method [3, 4, 5]. By first detecting the keyframes and then processing those frames as if they were images, this approach can also be utilized on video [1].

1.1 Purpose of this Document

This project aims to successfully detect tampering in any image or video and reconstruct them to their original state. We have made extensive use of Digital Image Processing (DIP) concepts in this project. This will be helpful in many areas, such as copyright infringement of content on social media. Moreover, this project will also be an essential asset in law enforcement agencies where CCTV footage is presented as evidence. Our project can help place watermarks on such footage and will successfully detect any tampering done by some third party to erase any concrete evidence from the footage. We have studied various watermarking algorithms from research papers mentioned in the reference section. However, the main watermarking algorithms used will include the use MD5 hashing function [5] to self-embed the watermark into an image.

Additionally, we have made use of this same methodology to watermark videos. The main idea is to identify the keyframes from a window of frames and watermark that frame according to the same algorithm used to watermark the image [1]. We will perform watermarking methods on pictures and videos to later be detected for any tampering through our system. In case of any tampering, the watermarked image or video will be recovered depending on the percentage of tampering.

1.2 Intended Audience

This document is intended for our team of developers, the FYP supervisor, the evaluation team, and the FYP coordinator. All of the people mentioned above are the document's intended audience because they are all involved in this project.

1.3 Definitions, Acronyms, and Abbreviations

CZ refers to the central cropping attack on an image (center of image). VCZ means a vertical center attack on an image. Square image refers to an n×n dimension image where n is any positive number. For example, 256×256 or 512×512 image dimensions have been used for the experimental setup. Image tampering includes cropping attacks and editing attacks on an image. Video tampering includes object removal attacks on video frames. Keyframes are selected from a window of frames with the most information of that specific window of frames.

2. General Description

2.1 User Characteristics

Most people who work with photos, videos, social media users, and law enforcement agencies will use our Self Embedding Watermarking System. Because editing tools are widely available on the internet and it is relatively simple to modify and alter a photo or video to suit one's needs, this can lead to problems and fake scandals, causing people's reputations to be ruined. This is especially true for well-known individuals, such as celebrities, who are also very active on social media. On the other hand, law enforcement agencies can use this system to determine whether evidence has been tampered with and, if desired, restore the video or image to its original state. So, using our system, users will be able to authenticate the video and image, and if it has been forged, they will be able to restore it.

2.2 Domain Overview

The Self Embedding Watermarking System has two main features: tamper detection and media restoration. The media, in this case, includes both images and videos. The system first allows the user to upload an image or video, after which the user can choose whether he only wants to use tamper detection, watermark an image or video, or restore an image or video that has been watermarked. To begin, if the user only wants to detect tampering in the media, a new tab is opened, and the result is displayed. Second, if the user wishes to watermark an image or video, the media is watermarked, and the user is given the option to download the watermarked media. Finally, if the user wishes to restore a watermarked media, he or she should upload it, and the system will open another tab with the option to download the restored media. So, in a nutshell, our system enables its users to authenticate a video or an image, watermark a video or an image, and finally, restore a video or an image to its original state.

3. Functionality

The functional requirements can be divided between the user and the watermarking system.

3.1 Functional Requirements

1. The system shall allow the user to place a watermark in an image or video.
2. The system shall allow the user to download watermarked images or videos.
3. The system shall allow the user to detect any tampering in watermarked images or videos.
4. The system shall allow the user to view detection results on tampered images or videos.
5. The system shall allow the user to reconstruct tampered images or videos that have been watermarked by the system.
6. The system shall allow the user to download reconstructed images or videos.
7. The system shall generate reports on tampered regions of watermarked images or videos.

3.2 Non-Functional Requirements

The following are non-functional requirements identified for the system:

3.2.1 Performance Requirements

The System shall meet the following performance requirements:

- The system shall limit the size of the uploaded images or videos to 50MB.
- The system webpage shall load in 2 seconds.
- The system shall take an approximate time of 180 seconds (3 minutes) to embed watermark in an image [5].
- The system shall take an approximate time of 180 seconds 180-199 seconds to recover tampered image depending on the the percentage of tampering (25% - 75%) [5].

3.2.2 Security Requirements

The system shall meet the following security requirements:

- The system shall run image or video processing algorithms on server side to maintain secrecy of watermarking and reconstruction algorithms.
- The system shall use a permutation key to place watermarks on images.

3.2.3 Usability Requirements

The following usability requirements shall be met by the system:

- The system shall provide an easy to learn and navigate user interface.
- A consistent theme shall be used throughout the system.

3.2.4 Sustainability

- The system shall be accessible from any computer browser assuming the hardware and software specifications mentioned in hardware and software requirements are met.

3.2.5 Reliability

- The system shall have a 100% uptime guarantee.

3.2.6 Extensibility

- The system shall be generic i.e., it shall work with most image or video formats.

3.3 Assumptions

The following assumptions have been taken into consideration while designing the specifications for this system:

- The user shall upload images or videos having size maximum of 50MB.
- The system server shall create an instance container for each user to lessen the load on the main server, however this will increase the cost of the server.
- The user shall upload an image having nxn dimensions i.e., a square image.
- The cost of the server shall increase proportionally to the number of users.

4. Use Cases

The Self Embedding Watermarking System has the following use cases which the users can use to perform various tasks

- Watermark Image
- Watermark Video
- Image Tamper Detection
- Video Tamper Detection
- Video Reconstruction
- Image Reconstruction

5. System Requirements

The project shall require the following hardware and software requirements:

5.1 Hardware Requirements

The hardware requirements for the usage of this system are the following:

- Cloud server for deployment of system. The server will be responsible for handling the complex computations of images or videos.
- A GPU, preferably GTX 1050 ti.
- CPU Intel Core i5-7500 (or higher) 3.40 GHz.
- 4GB RAM or higher.

5.2 Software Requirements

The software requirements for the usage of this system are the following:

- NodeJS
- ReactJS
- React-Bootstrap
- MATLAB

References

[1] V. Amanipour and S. Ghaemmaghami, "Video-Tampering Detection and Content Reconstruction via Self-Embedding," in IEEE Transactions on Instrumentation and Measurement, vol. 67, no. 3, pp. 505-515, March 2018, doi: 10.1109/TIM.2017.2777620.

[2] Q. Yang, D. Yu, Z. Zhang, Y. Yao and L. Chen, "Spatiotemporal Trident Networks: Detection and Localization of Object Removal Tampering in Video Passive Forensics," in IEEE Transactions on Circuits and Systems for Video Technology, doi: 10.1109/TCSVT.2020.3046240.

[3] Gul, E., Ozturk, S. A novel hash function based fragile watermarking method for image integrity. Multimed Tools Appl 78, 17701–17718 (2019). https://doi.org/10.1007/s11042-018-7084-0

[4] Gul, E., Ozturk, S. A novel pixel-wise authentication-based self-embedding fragile watermarking method. Multimedia Systems 27, 531–545 (2021). https://doi.org/10.1007/s00530-021-00751-3

[5] Gul, E., Ozturk, S. A novel triple recovery information embedding approach for self-embedded digital image watermarking. Multimed Tools Appl 79, 31239–31264 (2020). https://doi.org/10.1007/s11042-020-09548-4

Appendix

Appendix A: Time Complexity

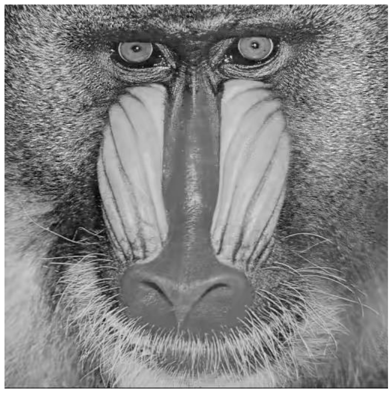
Table 1: Time Complexity for Watermarking

Following table shows time complexity for watermark embedding process

Image	Watermark embedding time (sec)		
	Recovery bits embedding	Authentication bits embedding	Total watermark embedding
Lena	133.128627	48.839553	179.968180
Sailboat	132.485935	46.644259	179.130194
Lake	132.879485	46.652739	179.532224
Airplane	132.357713	46.389058	178.746771
Baboon	133.314231	46.128864	179.443095
Goodhill	133.624994	46.151119	46.151119

Time complexity for watermark embedding is given in table 1. The images mentioned in the table are given below. All the mentioned images are square images, i.e., they have n×n dimensions. For the images mentioned in the table, the dimensions are 512×512. The computing time for watermark embedding depends on the size of the image. This is due to the whole image being divided into various blocks. These experiments have been conducted on a CPU Intel Core i5-7500 3.4 GHz with 4GB RAM and a GTX 1050 ti GPU [1]. Hence the processing time might be reduced depending on a higher specs CPU and GPU.

(a)



(b)

(c)

(d)

(e)

(f)

Figure 1: (a) Lake, (b) Baboon, (c) Lena, (d) Goodhill, (e) Sailboat, (f) Airplane

For the process of recovering these watermarked images, the time complexity varies according to the amount of tampering done in the image. The details are mentioned in table 2. Maximum limit for image recovery is 75% which takes the longest time to recover.

Table 2: Time Complexity for Recovering

Following table shows the time complexity for image recovering process

Image	Tamper detection and recovery time (sec)		
	25% CZ	50% VCZ	75% CZ
Lena	180.296284	189.142496	198.514490
Sailboat	180.821834	190.707774	199.018766
Lake	179.811014	189.824777	199.324460
Airplane	180.477713	188.998814	198.762606
Baboon	180.199268	189.820586	198.406937
Goodhill	180.330294	189.730310	189.730310