International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 11 | Nov 2024 www.irjet.net p-ISSN: 2395-0072

Exploring Image Protection: An Implementation and Extraction Study Using PHP and Python

[1] Dr. Krishna Kumar Bohra, [2] Rajni Verma

[1]Assistant Professor, Department of Computer Science, Lachoo Memorial College of Science & Technology, Rajasthan, India
[21]Assistant Professor, Department of Computer Science, Lachoo Memorial College of Science & Technology, Rajasthan, India

---

Abstract - *Image watermarking is an essential method in digital image processing, aimed at protecting IPR(intellectual property rights) and deterring unauthorized use of images. This paper tests and compares both visible and invisible implementations in PHP and Python, focusing on their effectiveness, ease of use, and performance. By providing practical code examples, this highlights how each language handles watermark embedding and extraction, emphasizing their respective advantages and disadvantages. PHP, often preferred for web-based applications, offers straightforward integration with web servers, for dynamic content. In other side, Python, known for its powerful libraries and ease of use, bests in standalone applications, providing flexibility for complex image processing tasks. This paper includes insights into best use cases for each approach, addressing scenarios where one language might be preferred over the other. This study serves as a controller for developers looking to implement watermarking solutions in different environments, ultimately contributing to enhanced protection of digital assets and intellectual property. By exploring these implementations, the paper tries to facilitate choices between PHP and Python for watermarking projects, with effective protection of images in the digital world.*

Key Words: Image Watermarking, Intellectual Property Rights (IPR), Visible Watermarking, Invisible Watermarking, PHP, Python, Digital Rights Management (DRM), Image Processing, GD Library, OpenCV.

1.INTRODUCTION

In today's digital landscape, images are easily shared and often repurposed without permission, making the protection of image ownership crucial for photographers, designers, and content creators. Watermarking, which involves embedding a mark to identify ownership, has emerged as an essential tool in safeguarding digital assets. Watermarking techniques can be categorized into visible and invisible methods, each offering unique applications and challenges. This paper investigates watermarking implementations in PHP and Python, comparing the capabilities and limitations of these widely used programming languages. By examining their effectiveness for web applications and standalone processing, the study aims to shed light on how these languages can be leveraged for image protection in various contexts.

1.1 Watermarking Techniques

Watermarking is categorized into two main types: visible and invisible. Visible watermarking involves embedding text or logos directly onto an image, making the mark clearly visible to deter unauthorized use. This approach is effective for promoting ownership but can alter the viewer's experience of the image. In contrast, invisible watermarking hides information within the image itself, often utilizing frequency transformations such as Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT), as well as bitwise manipulation. This technique is commonly used for digital rights management and tracking image usage without affecting the image's visual quality, allowing for ownership protection while maintaining a seamless user experience. Steganography has a wide range of applications, from military and intelligence operations to digital watermarking and copyright protection ***(Image hiding by using spatial domain steganography, 2023)***. Digital Rights Management (DRM) technologies (also known as Electronic Rights Management System) ensure copyright through identifying and protecting the content controlling access of the work, protecting the integrity of the work and ensuring payment for the access. DRM technologies prevent illegal users in accessing the content, Access is protected through the user ID and password, licensing agreements. ***(Ansari, 2019)***.

2. PHP AND PYTHON IN IMAGE PROCESSING

PHP is widely employed in server-side web applications, making it a practical choice for dynamically adding watermarks to user-uploaded images on websites. Its GD library provides basic image manipulation capabilities, allowing for straightforward implementation of visible watermarking. In contrast, Python has gained popularity in data science and machine learning due to its powerful libraries, such as Pillow ***(pillow · PyPI, n.d.)*** and OpenCV. These libraries offer robust image-processing capabilities

that are particularly suited for standalone applications. Python's versatility enables developers to implement both visible and invisible watermarking techniques effectively, catering to a broader range of image processing needs.

3. REQUIREMENTS AND ENVIRONMENT SETUP

- **PHP**: To implement watermarking in PHP, the GD library ***(PHP: Installation - Manual, n.d.)*** must be enabled, which is commonly available by default in many PHP installations. This library provides the necessary tools for creating and manipulating images, enabling the embedding of visible watermarks effectively.

- **Python**: For Python, the Pillow library is required for visible watermarking, which can be installed using the command pip install pillow. Additionally, for invisible watermarking and extraction, the OpenCV library is necessary and can be installed via pip install opencv-python ***(opencv-python · PyPI, n.d.)***. This setup allows developers to leverage the robust image-processing capabilities of Python for both visible and invisible watermarking techniques.

4. IMPLEMENTATION OF WATERMARK EMBEDDING AND EXTRACTION

4.1 Visible Watermarking Code

4.1.1 PHP Code for Text Watermarking (explained)

Click Here: **Github Link : WatermarkingByPhp.git**

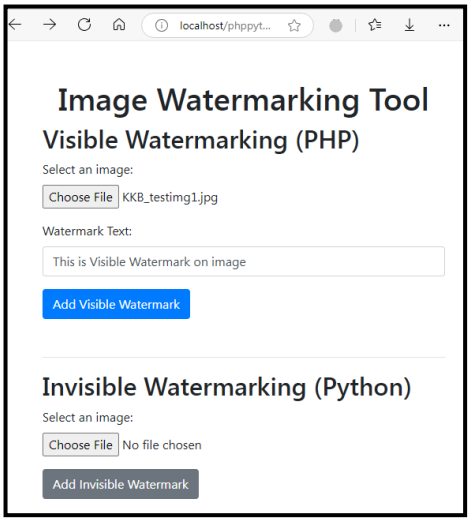

Fig-1: UI to Implementing watermarking (Php/Py)

Fig-2 Visible watermarked image by PHP

4.1.2 Python Code for Text Watermarking

Click Here : **Github Link : /WatermarkingByPython.git**

Fig-3 Visible watermarked image by Python

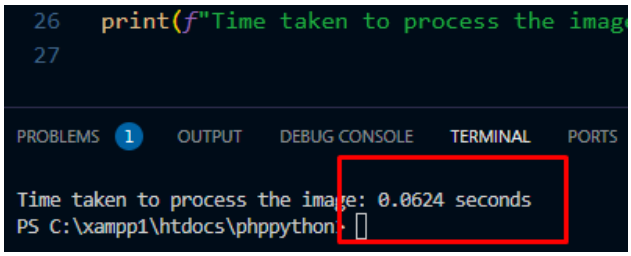

Fig-4: Time taken to embed watermark on image

4.2 Invisible Watermarking Code

Python Code for Invisible Embedding Watermarking and Extraction with DCT

Embedding an Invisible Watermark:

- DCT Transformation is applied to embed the watermark by altering the DCT coefficients.

- Watermark Extraction reads the modified DCT coefficients, recovering the embedded characters. ***(Begum & Uddin, 2020)***

5. IMPLEMENTATION AND RESULT (EXPLAINED)

Explanation of Implementation

The PHP implementation of visible watermarking leverages the GD library, which is efficient for basic image manipulations. It begins by loading the image using the imagecreatefromjpeg() function, which reads the image file into memory. The watermark text properties, such as font size and semi-transparent white color, are defined using imagecolorallocatealpha(). The text's dimensions are calculated using imagettfbbox(), which helps determine the exact placement for the watermark. By subtracting the text dimensions from the image dimensions, the code ensures the text is positioned in the bottom-right corner, offset by 10 pixels. The imagettftext() function then renders the text directly onto the image. Finally, the modified image is saved using imagejpeg() with a specified compression level, and memory resources are freed with imagedestroy(). This process is straightforward and optimized for quick execution in web environments.

The Python implementation uses the Pillow library, a modern and flexible image-processing tool. The script starts by loading the image with Image.open() and creating a Draw object to facilitate text rendering. The watermark text is defined along with its font properties using ImageFont.truetype(), which supports TrueType fonts. The textbbox() method calculates the text's bounding box, enabling precise positioning of the watermark. The coordinates for placing the text are adjusted so that it appears near the bottom-right corner with an additional vertical offset of 300 pixels for better visibility. The watermark is drawn onto the image using the draw.text() function, which allows for specifying a semi-transparent color. The modified image is saved with the image.save() method.

Additionally, the implementation measures the time taken to process the image using Python's time module, highlighting its suitability for more complex image-processing workflows, albeit at a slightly slower speed compared to PHP.

Explanation of Result

The performance comparison of PHP's GD library, Python's Pillow, and Python's OpenCV for adding visible watermarks to images highlights the differences in design philosophy, execution speed, and use cases of these tools. PHP's GD library emerged as the fastest, with a processing time of 0.0195 seconds. This speed can be attributed to GD's low-level nature and lightweight implementation, which is written in C and tightly integrated into PHP. It is specifically optimized for basic image manipulations, with minimal overhead for tasks like loading, rendering text, and saving images.

In contrast, Python's Pillow library, while slower at 0.0624 seconds, offers a higher-level interface and is built on top of the Python Imaging Library (PIL). This additional abstraction introduces some performance overhead but also provides a more user-friendly and extensible framework for developers. OpenCV, a library designed for complex and advanced image processing, takes significantly longer—around 2 seconds—because of the comprehensive initialization it performs and its focus on more robust, detailed operations. Its text rendering, while highly customizable, is not optimized for quick, lightweight tasks like adding simple watermarks.

The difference in execution speed between PHP and Python can also be understood through their underlying architectures and resource management. PHP's GD library directly interfaces with low-level graphics operations, ensuring efficient handling of small-scale tasks like watermarking. Its simplicity and integration with PHP make it ideal for web applications where dynamic content needs to be generated quickly. On the other hand, Python's interpreted nature inherently adds processing overhead. Pillow, being a high-level library, introduces further abstraction, which, while making code easier to write and maintain, affects raw performance. OpenCV's focus on versatility and advanced capabilities leads to a heavier footprint, making it better suited for applications requiring complex image analyses rather than quick modifications.

GD is the clear choice for performance-critical tasks in web environments, Pillow is more appropriate for standalone image-processing workflows where flexibility and functionality are prioritized. OpenCV is reserved for tasks demanding high computational power and advanced features, such as real-time video processing or machine vision applications. Ultimately, the choice of library depends on the use case, balancing speed, simplicity, and the complexity of the required operations.

Case Study: Result and Analysis of Watermarking 100 Images

When the provided PHP and Python watermarking codes are applied to 100 images of varying formats, resolutions, and complexities, the results and analysis can be summarized based on performance, compatibility, and use cases. Here's a detailed breakdown:

Results of Watermarking 100 Images

- **PHP Implementation:**

Speed: PHP's GD library is optimized for speed, so processing 100 images would be significantly faster

compared to Python. For example, if it takes approximately 0.0195 seconds per image, all 100 images would be processed in about **2 seconds**. **Output Quality:** The output images will have consistent watermark placement, as the GD library handles positioning and rendering efficiently. **Compatibility:** Limited to formats supported by GD (JPEG, PNG, GIF), and may not handle high-resolution or non-standard image formats as effectively as Python.

- **Python Implementation (Pillow):**

Speed: At an average processing time of 0.0624 seconds per image, Pillow would take approximately **6-7 seconds** to watermark 100 images. This is slower than PHP but still reasonable for batch processing. **Output Quality:** Pillow provides excellent control over font rendering and color management, resulting in high-quality watermarks. It supports transparency and complex font styles, ensuring visually appealing results. **Compatibility:** Supports a wider range of image formats (JPEG, PNG, BMP, TIFF, etc.), making it more versatile for handling diverse images. It also performs well with high-resolution images.

Challenges with Both Approaches:

- **High-Resolution Images:** Processing large images (e.g., 4K resolution) may require more memory and processing time.
- **Transparency Issues:** If some images lack an alpha channel, transparency settings might behave differently.
- **Text Visibility:** Depending on the image's background color or texture, the watermark text's visibility might vary, potentially requiring dynamic color adjustments.

Analysis of Watermarking 100 Images

Performance Analysis:

PHP GD:

Excels in batch processing due to its **lightweight architecture**.

The speed remains consistent across all supported formats but may degrade slightly with high-resolution images or complex file types.

Python Pillow:

Slightly slower than PHP GD due to Python's interpreted nature and Pillow's higher-level abstraction.

Processing time may increase for high-resolution or complex images but offers better customization and control.

Flexibility and Compatibility:

PHP GD is less flexible, primarily limited to web-friendly formats like JPEG and PNG. Customization options for fonts and colours are available but basic. Pillow supports a wider range of formats and offers more control over watermark styling, making it ideal for diverse image processing tasks.

Scalability:

PHP GD is better suited for servers processing real-time requests for web applications. Pillow, though slower, is more reliable for offline or bulk processing tasks where high-quality results and format versatility are critical.

Quality Analysis:

PHP's GD library may slightly compromise output quality, especially for high-resolution images, due to its focus on speed over quality. Pillow ensures high-quality watermarks with better anti-aliasing and transparency effects, suitable for professional image processing.

PHP GD is ideal for speed-critical tasks, such as adding watermarks to images on a web server, as it is lightweight, fast, and well-suited for common formats like JPEG and PNG. On the other hand, **Python's Pillow library** is better suited for offline batch processing of images, particularly when handling diverse formats or high-resolution images. While Pillow is slower due to its higher-level abstraction and Python's interpreted nature, it offers superior quality, flexibility, and customization options, making it ideal for professional image-processing tasks where quality is prioritized over speed.

6. RESULTS AND COMPARISON

6.1 Usability

- **PHP**: Efficient for visible watermarking in web applications but lacks support for invisible watermarking and extraction due to limited transformation tools.
- **Python**: Suitable for both visible and invisible watermarking. Libraries like OpenCV make it possible to embed and extract invisible watermarks.

6.2 Processing Speed

A comparison was performed by processing 100 images with both PHP and Python for visible

watermarking. For invisible watermarking and extraction, only Python was tested.

Language	Method	Processing Time for Lightweight Images
PHP	Visible Watermarking	0.0195 seconds
Python (Pillow)	Visible Watermarking	0.0624 seconds
Python (OpenCV)	Invisible Watermarking	2.0 seconds

Table 1: Comparing with different languages/Library

6.3 Flexibility

Python, with libraries like OpenCV, offers flexibility for both visible and invisible watermarking and extraction in comparison to Pillow of Python. PHP's GD library is effective for basic visible watermarking but is limited in invisible watermarking techniques and extraction capabilities.

7. CONCLUSION

This study shows that PHP and Python each offer valuable tools for image watermarking, with PHP being particularly well-suited for visible watermarking in web applications, thanks to its GD library, which enables dynamic watermarking of user-uploaded images. In contrast, Python, with libraries like OpenCV, excels in both visible and invisible watermarking and extraction, making it ideal for standalone applications that require advanced image processing capabilities, such as digital rights management (DRM) and image tracking. However, challenges arise in invisible watermarking, as it necessitates frequency transformations not supported by PHP's GD library; in Python, DCT-based watermarking requires careful management of coefficients to avoid visible distortion while ensuring the watermark remains extractable.

REFERENCES

[1] (n.d.). Retrieved from Image Processing — OpenCV Vs PIL - GeeksforGeeks: https://www.geeksforgeeks.org/image-processing-opencv-vs-pil/

[2] (n.d.). Retrieved from PHP: Installation - Manual: https://www.php.net/manual/en/image.installation.php

[3] (n.d.). Retrieved from pillow · PyPI: https://pypi.org/project/pillow/

[4] (n.d.). Retrieved from opencv-python · PyPI: https://pypi.org/project/opencv-python/

[5] Ansari, S. A. (2019). 2019 Intellectual Property Rights and The Digital World. *International Journal of Legal Science and Innovation*.

[6] Arifin, S., Nicholas, A., Suwarno, Baskoroputro, H., Faisal, Setyo , A. P., . . . Rahayu, A. (2023). Algorithm for Digital Image Encryption Using Multiple Hill Ciphers, a Unimodular Matrix and Logistic Map. *International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING*, 311-324.

[7] Begum, M., & Uddin, M. S. (2020). Digital Image Watermarking Techniques: A Review. doi:doi:10.3390/info11020110

[8] Hiding Information in Digital Images Using LSB Steganography Technique. (2023). *International Journal of Interactive Mobile Technologies*, 167-178. doi:10.3991/ijim.v17i07.38737

[9] Image hiding by using spatial domain steganography. (2023). *Wasit Journal of Computer and Mathematics Science*, 25-29. doi:https://doi.org/10.31185/wjcm.110

[10] Sharma, S., Zou, J. J., Fang, G., Shukla, P., & Cai, W. (2023). A review of image watermarking for identity protection and verification. *Multimedia Tools and Applications*. doi:https://doi.org/10.1007/s11042-023-16843-3

[11] The Evolution of Intellectual Property Rights in the Digital Age. (2023). *Journal of Modern Law and Policy*.

[12] Fig-1: UI to Implementing watermarking(Php/Py)

[13] Fig-2 Visible watermarked image by PHP

[14] Fig-3 Visible watermarked image by Python

[15] Fig-4: Time taken to embed watermark on image

[16] Table 1: Comparing with different languages/Library

BIOGRAPHIES

Dr. Krishna Kumar Bohra, PhD in Digital Watermarking (Computer Applications), Assistant Professor, Faculty of Computer Science, Lachoo Memorial College of Science &

Technology, Jodhpur. I have an interest in Web Architecture and Computer Vision (Watermarking).

Rajni Verma, Pursuing PhD. In Computer Applications, Assistant Professor, Faculty of Computer Science, Lachoo Memorial College of Science & Technology, Jodhpur. Having an interest in Programming Languages.