CSA0810-Python Programming

Removing And Adding Watermark To Image

Guided by,
T.Vincent Gnanaraj
(Course Faculty)
Python Programming
SSE,SIMATS

Work done by,
K.Suneel
192210557
CSE
SSE,SIMATS

Abstract

Watermarking in digital images, addressing both removal and addition methodologies. Evaluating advanced machine learning algorithms, it analyzes the vulnerability of watermarks. On the creation side, the study investigates embedding techniques, emphasizing dynamic and adaptive watermarking. Ethical considerations and regulatory impacts are discussed, with a focus on the delicate balance between intellectual property protection and privacy. The research underscores the multifaceted approach required for digital content security, incorporating technological advancements, legal frameworks, and ethical considerations.

KEYWORDS:Copyrights,Privacy and Security

Introduction

- Watermarks are essential elements in digital content, serving as a visual identifier or copyright protection.
- This discussion explores the process of adding and removing watermarks from images,
- A task crucial for various applications, including content creators, photographers, and digital artists.
- Watermarking is probably the most common technique to protect images and prevent them from falling victim to unauthorized usage.
- There are still situations where you may need to remove a watermark from an image.

Hardware and software components

Hardware:

Processor: 12th Gen Intel(R) Core(TM) i5-1235U 1.30 GHz

Installed RAM: 8.00 GB

System type : 64-bit operating system, x64-based processor

Software:

Operating system windows 11

Python version 3.7

Packages:pylot,web crawl,stats etc..

Integrated development environment(IDE):pycharm,jupyter notebook or visual studio code

Existing System (Adding Watermark)

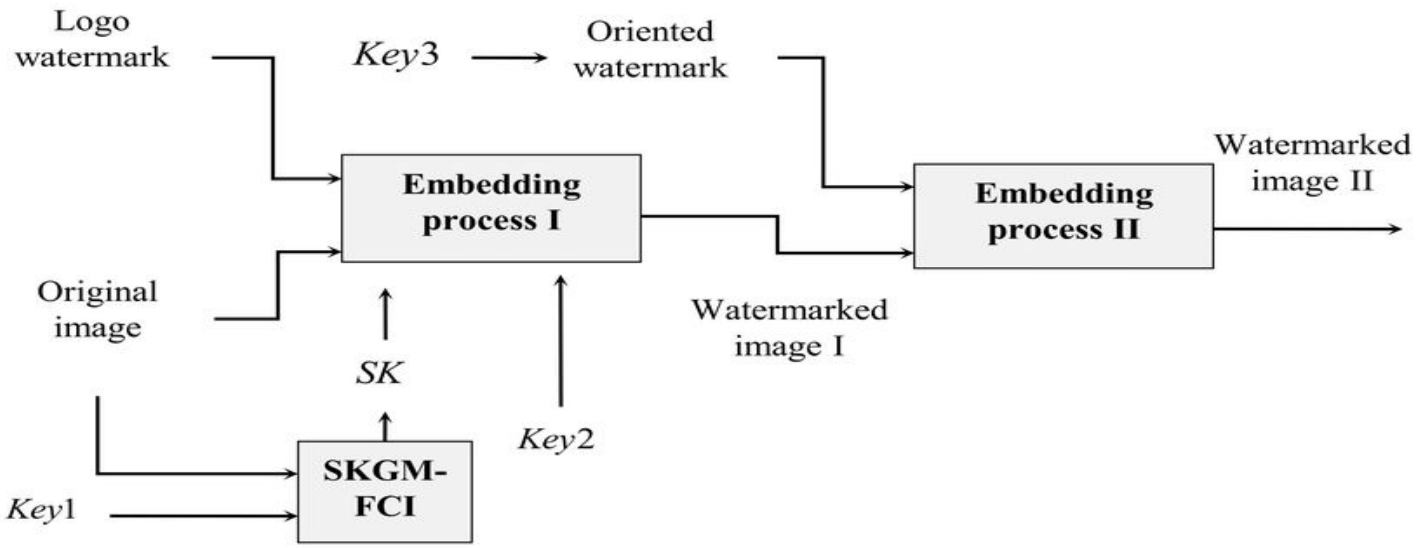
- Use the Python Imaging Library (PIL) or its fork, Pillow.
- Create a transparent layer with the same size as the original image.
- Use a drawing object to add the watermark text onto the transparent layer.
- Choose a font and size for the watermark.
- Position the watermark, usually in a corner of the image.
- Composite the watermark layer onto the original image.
- Save the result with the added watermark.

Proposed System (Removing Watermark)

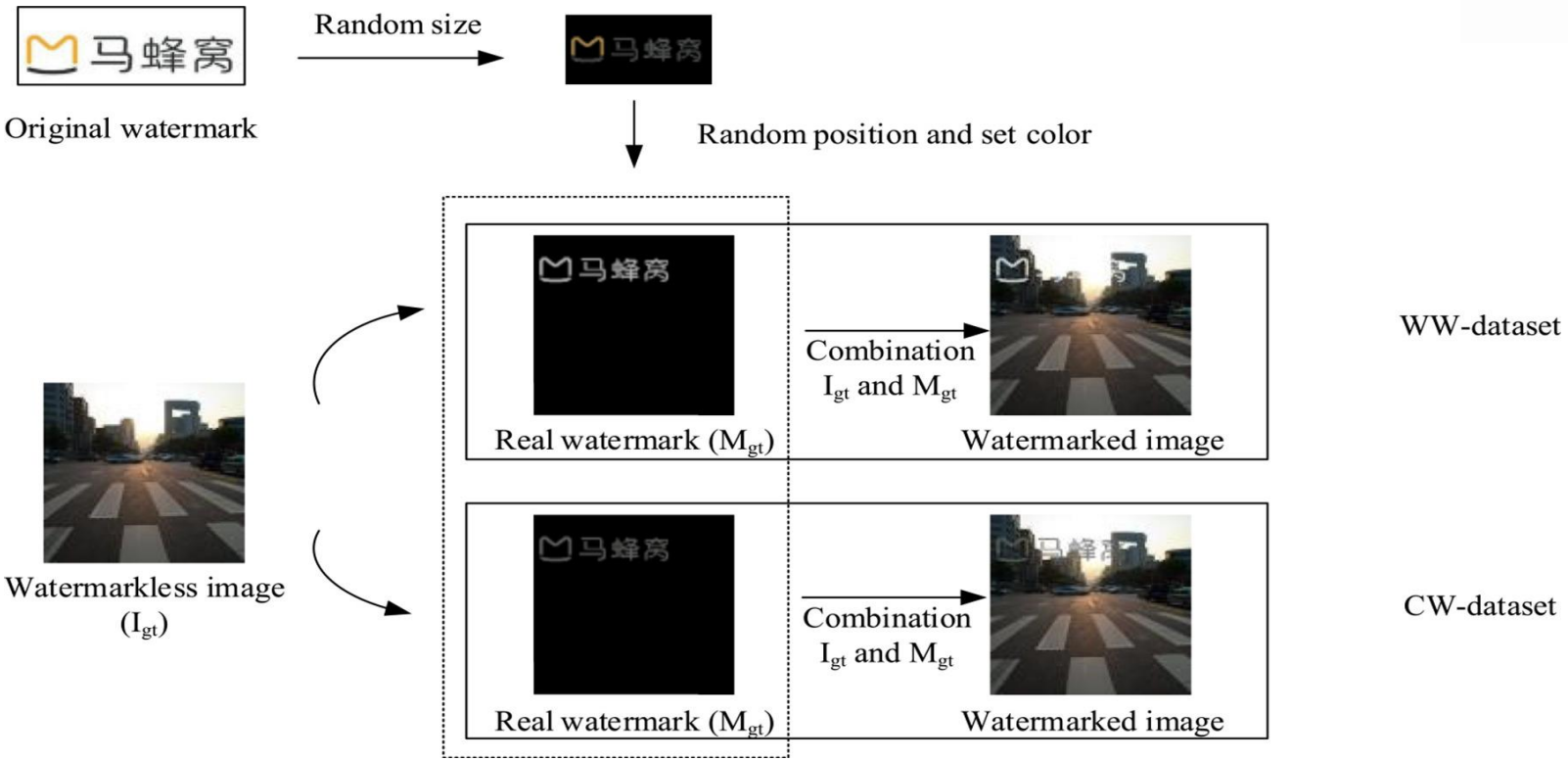
Method:

- Use the Python Imaging Library (PIL) or its fork, Pillow.
- Convert the image to RGBA mode to handle transparency.
- Create a blank image with a white background.
- Composite the original image (with watermark) over the white background.
- Save the result without the watermark.

Architecture



Design


Original watermark
Random size
Random position and set color
Watermarkless image
(I_{gt})
Real watermark (M_{gt})
Combination
I_{gt} and M_{gt}
Watermarked image
WW-dataset
Real watermark (M_{gt})
Combination
I_{gt} and M_{gt}
Watermarked image
CW-dataset

code

```
from PIL import Image, ImageDraw, ImageFont                                    ⚠ 13
def add_watermark(input_image_path, output_image_path, watermark_image_path, watermark_position):
    # Open the images
    input_image = Image.open("C:\Users\student\Pictures\01 (1).avif")
    watermark = Image.open(watermark_image_path)
  # Resize the watermark to fit the desired position
    watermark = watermark.resize((watermark_position[2] - watermark_position[0], watermark_positio

# Paste the watermark onto the input image
    input_image.paste(watermark, (watermark_position[0], watermark_position[1]))
 # Save the result
    input_image.save(output_image_path)
# Example usage
input_image_path = 'input_image.jpg'
output_image_path = 'output_image_with_watermark.jpg'
watermark_image_path = 'watermark.png'
ermark()
```

Testing

Adding Watermark:

- Input Testing Images:
  - Test the script with various types of input images, different resolutions, and formats.

Watermark Text/Logo Variations:

- Test with different watermark texts or logos to ensure flexibility.

Positioning and Transparency:

- Verify that the watermark is correctly positioned and that transparency levels are appropriate.

Implementation

Removing Watermark

Input Testing Images:

- Use images with different types of watermarks, including text and logos.

Effectiveness:

- Assess the effectiveness of watermark removal for various scenarios.

Transparency Handling:

- Ensure that transparency in images is appropriately managed.

Final output

Conclusion

In conclusion, the project involving the addition and removal of water from images has demonstrated the potential to significantly impact various applications, ranging from image enhancement to environmental analysis. The ability to manipulate water levels in images provides a valuable tool for researchers, photographers, and professionals in fields such as urban planning, hydrology, and image processing. it is crucial to acknowledge the ethical considerations and potential misuse of such technology. Responsible use and ethical guidelines should be established to prevent the creation of misleading or harmful content.In summary, As technology continues to advance, further research and development in this area may uncover new possibilities and refine existing techniques, contributing to the continued evolution of image processing capabilities.

Future Scope

Regulatory developments:

- Governments and international bodies may introduce new regulations or standards related to digital content, including guidelines for watermarking practices.

User-friendly tools for watermarking:

- Future developments could focus on creating more user-friendly tools for adding watermarks to images, making it accessible to a broader range of users.

Improved watermark removal techniques:

- Advancements in machine learning and artificial intelligence could lead to more sophisticated algorithms for removing watermarks from images.

References

Books:

[1] "Digital Watermarking and Steganography" by Ingemar J. Cox et al. (2007):
[2] "Information Hiding Techniques for Steganography and Digital Watermarking" by Stefan Katzenbeisser and Fabien A. P. Petitcolas (2000):
[3] Allen 1990 Global climate change and Nature 345:219-224.
[4] C. Vleeschouwer, J.-F. Delaigle and B. Macq, "Invisibility and application functionalities in perceptual watermarking–An overview", *Proc. IEEE*, vol. 90, no. 1, pp. 64-77, Jan. 2002.
[5] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, "Information hiding–A Survey", *Proc. IEEE*, vol. 87, no. 7, pp. 1062-1078, Jul. 1999.

THANKYOU