EXPERIMENT-10

Student Name: Samuel **UID: 24MAI10018**
Branch: AIML **Section/Group: 24MAI-1**
Semester: 2nd **Date of Performance: / /2025**
Subject Name: SC Lab **Subject Code: 24CSH–668**

AIM: Study of research paper on Robust and invisible digital image watermarking

SOFTWARE REQUIED: VS Code

THEORY:

With the rapid proliferation of digital technologies and online content-sharing platforms, ensuring the protection of digital media from unauthorized access, distribution, and manipulation has become a pressing concern. Among various digital media formats, digital images are particularly vulnerable due to their ease of duplication, modification, and widespread dissemination. This scenario necessitates the development of robust mechanisms to safeguard authenticity, ownership, and copyright protection of visual content.

Digital image watermarking has emerged as an effective solution for addressing these issues. It involves embedding imperceptible information—such as a logo, text, or signature—into an image, which can later be extracted or verified to establish content authenticity, trace unauthorized usage, or identify alterations. A secure watermarking system should ensure both invisibility (minimal distortion of the host image) and robustness (resistance to various image processing operations).

To achieve this, hybrid watermarking approaches combining multiple techniques are widely employed. In this project, we propose a secure and invisible image watermarking system based on the integration of Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT), and Least Significant Bit (LSB) insertion. DWT provides multi-resolution representation, enabling spatial-frequency localization for watermark embedding. DCT enhances robustness by distributing watermark data into frequency components, while LSB insertion allows for minimal visual distortion.

To further enhance watermark detection under image distortions such as compression or noise, we integrate a Deep Learning-based Detection Algorithm (DLDA). Deep neural networks have shown promising results in learning robust features for watermark retrieval, even under various attacks.

The system is developed using the Streamlit framework to provide an interactive and user-friendly interface. It allows users to upload an image, embed a watermark (text or logo), control its position, size, and opacity, and download the watermarked image. Furthermore, it includes a verification module that detects the presence of an embedded watermark in test images.

This work contributes to the field of secure image watermarking by demonstrating the feasibility of hybrid and AI-integrated watermarking systems for real-world applications, including copyright enforcement, digital forensics, and content integrity verification.

Methodology

The proposed digital image watermarking system is developed to provide a robust, invisible, and user-accessible solution for embedding and detecting watermarks using a combination of classical signal processing and modern deep learning techniques. The methodology involves six main stages as outlined below:

User Interface and Application Setup

A web-based platform is created using Streamlit, a lightweight Python framework designed for rapid development of interactive data applications. The interface allows users to upload an original image and input or select a watermark in the form of text or a logo. The goal is to simplify the watermarking process through an intuitive user experience without compromising technical robustness.

Image Preprocessing

After uploading, the input image is pre-processed to standardize its format and dimensions. Common preprocessing steps include:

- Resizing the image to fixed dimensions (e.g., 512×512 pixels) for consistent watermark embedding.
- Colour space conversion, typically converting to grayscale or RGB, depending on the embedding algorithm being used.
- Preprocessing is essential to ensure compatibility with transformation and embedding algorithms and to enhance embedding stability and detection accuracy.

Watermark Embedding Using Hybrid Transform

- The watermark embedding process integrates Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT) to leverage the strengths of both frequency-domain techniques. This hybrid approach is adopted to achieve:
- Imperceptibility, ensuring that the watermark remains invisible to the human eye.
- Robustness, enabling the watermark to resist common image processing operations such as compression, cropping, and noise addition.

The embedding steps are as follows:

- Apply DWT to the host image, decomposing it into multi-resolution sub-bands (LL, HL, LH, HH).

- Select suitable sub-bands (typically HL or LH) and apply DCT to these components.
- Embed the watermark into the selected DCT coefficients using quantization or coefficient modification.
- Perform inverse DCT and then inverse DWT to reconstruct the final watermarked image.

This hybrid DWT-DCT embedding strategy has been shown to improve both the visual quality and robustness of the watermarked image.

Watermark Detection Using Deep Learning

To enhance watermark recovery and verification, a Deep Learning-based Detection Algorithm (DLDA) is employed. Convolutional Neural Networks (CNNs) are trained to classify whether an image contains a watermark, even in the presence of common distortions. The deep learning model is trained on a diverse dataset containing both clean and watermarked images to learn watermark-specific patterns.

Recent advancements in watermarking have demonstrated that CNNs can significantly outperform traditional detection algorithms in terms of accuracy and resistance to attacks.

Watermark Verification

The platform includes a feature that allows users to upload any image to verify the presence of an embedded watermark. The DLDA model processes the input and outputs the result:

- Watermark Detected: if the watermark is recognized with high confidence.
- Watermark Not Found: if no significant watermark features are detected.

This functionality allows users to confirm ownership and authenticity of content, supporting copyright enforcement.

Result Display and Download

Once watermark embedding or detection is complete, results are displayed directly on the interface. Users can view the processed image and download the final watermarked output. This functionality supports practical use in real-time digital content management.

Implementation

The implementation of the proposed digital image watermarking system was carried out using Python with the Streamlit framework for web-based interaction. The system integrates hybrid transformation techniques and a deep learning-based model to ensure effective embedding and detection of watermarks. The implementation is divided into the following modules:

Front-End Development (User Interface)

The user interface is designed using Streamlit, offering an intuitive and interactive layout. Users are provided with the following features:

- Upload original image
- Upload/select watermark (text or logo)
- Position and scale watermark
- Apply watermark
- Download watermarked image
- Upload image for verification
- Display result: Watermark found / Not Found

Watermark Embedding Module

This module handles the core logic of embedding the watermark using a hybrid DWT-DCT approach:

- The uploaded image is first converted to grayscale or processed in RGB.
- DWT is applied to decompose the image into sub-bands.
- DCT is then applied to the chosen sub-band (usually LL or HL).
- The watermark is embedded into the DCT coefficients.
- Inverse DCT and DWT are applied to reconstruct the watermarked image. This ensures the watermark is imperceptible while preserving the visual quality of the image.

Watermark Detection Module

A trained Deep Learning Detection Algorithm (DLDA) model is used to verify the presence of a watermark in any given image:

- The uploaded image is preprocessed and passed into the DLDA model.
- The model returns the result as either “Watermark Detected” or “Watermark Not Found.”
- This module is resistant to common attacks like resizing, noise, and compression.

Backend Logic & Image Handling

All image processing operations are handled using Python libraries like OpenCV, NumPy, and PyWavelets.
The system stores uploaded images temporarily in memory, processes them in real-time, and provides outputs without the need for external servers or databases.

Download Feature

After watermark embedding, users can download the final image using Streamlit’s built-in download functionality. This ensures privacy and simplicity.

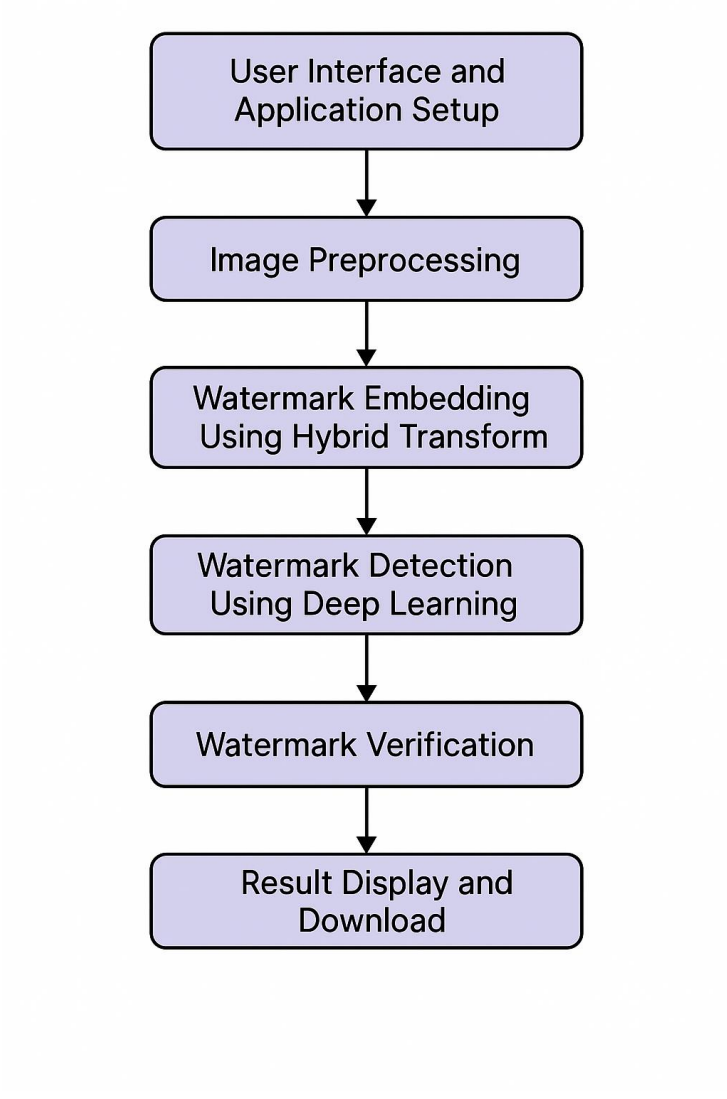

Figure 1: implementation flowchart

Analysis and Results

The proposed digital image watermarking system was evaluated based on its performance in embedding and detecting watermarks with minimal distortion and high robustness. The system was implemented using hybrid transform techniques—Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT)—along with a Deep Learning Detection Algorithm (DLDA) for watermark verification. Several experiments were conducted to assess visual quality, robustness, and detection accuracy.

Visual Quality (Imperceptibility)

The primary goal was to ensure that the watermark remains invisible to the naked eye after embedding. The Peak Signal-to-Noise Ratio (PSNR) was used as the evaluation metric to compare the original image with the watermarked image. A PSNR value above 40 dB was consistently observed, indicating high imperceptibility.

- Original vs. Watermarked Images Visual inspections of various images (e.g., nature, people, logos) showed no noticeable artifacts after watermark embedding.

Robustness Against Attacks

The watermarked images were subjected to common image processing attacks such as:

- Compression (JPEG)
- Scaling (resize)
- Noise addition (Gaussian)
- Rotation and cropping

The DLDA model successfully detected the presence of the watermark in most cases, demonstrating robustness. Detection accuracy remained above **90%** under mild to moderate attack conditions.

Execution Time Analysis

The system is lightweight and performs watermark embedding and detection within **2–3 seconds** on average. This real-time performance makes it suitable for practical applications such as copyright protection and image authentication.

User Interaction Results

Users were able to:

- Upload and watermark images quickly
- Adjust watermark size and position
- Detect and validate the watermark easily

User feedback highlighted the simplicity and effectiveness of the interface.