Chapter 1
Introduction

1.1 Introduction to Digital Video Watermarking

The digital era has ushered in unprecedented advancements in the creation, distribution, and consumption of multimedia content. Among the various forms of digital media, video has emerged as a dominant medium due to its capacity to convey detailed information through a combination of visual and auditory elements. However, the widespread use of the internet for video transmission and distribution carries some problems, specifically regarding copyright protection and intellectual property rights.

Digital watermarking has become an essential tool to address these concerns. By embedding secret information within a video, such as a logo, text, or unique identifier, digital watermarking ensures that ownership can be proven while maintaining the video's quality. This process offers a robust mechanism to safeguard against unauthorized use, piracy, and tampering, making it indispensable in today's digital landscape.

What is Digital Watermarking?

Digital watermarking is a sophisticated technique of embedding additional information, frequently known as "watermark," within multimedia content such as images, audio, and video. This embedded data remains imperceptible to casual viewers while being detectable and extractable using specialized algorithms. Unlike other data-hiding techniques like steganography, the primary goal of watermarking is not secrecy but robustness and authenticity. This means that the watermark should survive various transformations such as compression, resizing, or noise addition.

The watermark can serve various purposes, including:

1. **Proof of Ownership**: Embedding creator identifiers to establish intellectual property rights.
2. **Authentication**: Ensuring that the content remains unchanged since embedding of the watermark into the source.
3. **Broadcast Monitoring**: Tracking the use of multimedia in broadcast channels.
4. **Forensic Tracking**: Identifying sources of unauthorized distribution.

1.2 Domains of Watermarking: Spatial vs. Transform

Watermarking can be understood by two broad domain classifications viz: **spatial domain** and **transform domain**.

1. **Spatial Domain Techniques**:
    - These methods involve directly altering pixel values of an image/frame for watermark embedding.
    - **Advantages**:
        - High perceptibility; watermarked frames appear visually identical to the original.
        - Easy implementation and lower computational requirements.
    - **Limitations**:
        - Susceptible to attacks like noice additions, compression, and geometric distortions.
        - Limited robustness compared to transform domain techniques.
2. **Transform Domain Techniques**:
    - Transform domain methods embed the watermark by manipulating coefficients in frequency or transform spaces such as Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), or Singular Value Decomposition (SVD).
    - **Advantages**:
        - Higher robustness against compression and geometric attacks.
        - Greater resilience to noise addition and resampling.
    - **Limitations**:
        - Higher computational complexity.
        - Moderate imperceptibility depending on the embedding strength.

1.3 Key Requirements of Effective Watermarking

An effective watermarking technique should balance the following essential properties:

1. **Robustness**:
    - The watermark must withstand various intentional and unintentional attacks, such as noise, compression, scaling, and rotation.
    - Metrics like correlation and bit error rate (BER) are used to measure robustness.

2. **Imperceptibility**:
    - The visual and auditory excellence of the watermarked video should remain indistinguishable from the original. Metrics such as Peak Signal-to-Noise Ratio (PSNR) and Mean Square Error (MSE) are used to estimate this property.

3. **Payload Capacity**:
    - The total data that can be embedded without compromising the quality or robustness of the watermark.
4. **Security**:
    - The watermark should not be detected, altered or removed by any of the Unauthorized users without specific keys or algorithms.
5. **Computational Efficiency**:
    - The algorithm should be optimized for real-time embedding and extraction processes.

1.4 Challenges and Common Attacks

Watermarked videos are often subjected to various transformations that may degrade the watermark's quality. Common challenges include:

1. **Compression**: Formats like MPEG and H.264 are used to spread on lossy compression that may degrade embedded watermarks.
2. **Noise Addition**: Random noise may obscure or distort the watermark.
3. **Geometric Attacks**: Cropping, rotation, and scaling can misalign the watermark, making extraction difficult.
4. **Protocol Attacks**: These aim to weaken the link between the watermark and the host video, such as re-watermarking or tampering.

1.5 Technological Advancements in Watermarking

Recent advancements in watermarking leverage hybrid techniques combining multiple transforms to achieve optimal robustness and imperceptibility. Examples include:

1. **Color Space Transformation**:
    - Using YCbCr or YUV color spaces to embed watermarks in the luminance (Y) channel, where our eye is more sensitive.

      $Y = 16 + 128.553*G + 65.481*R + 24.966*B$

      $Cb = 128 – 74.203*G – 37.797*R + 112*B$

      $Cr = 128 – 93.786*G + 112*R – 18.214*B$

      $R = Y + 0 * Cb + 1.402 * Cr$

      $G = Y - 0.344136 * Cb - 0.714136 * Cr$

      $B = Y + 1.772 * Cb + 0 * Cr$

2. **Hybrid Methods**:
    - Combining DCT with SVD, or DWT with DCT, to enhance resilience against attacks.
3. **Dynamic Thresholding**:
    - Adaptive algorithms adjust embedding strength based on content sensitivity.
4. **Machine Learning**:
    - Utilizing neural networks to improve watermark detection and resistance to emerging threats.

1.6 Applications of Digital Watermarking

The versatility of watermarking extends across various domains, including:

1. **Media and Entertainment**: Protecting movies, TV shows, and live broadcasts from piracy.
2. **Healthcare**: Confidentially inserting the information of patients in the medical reports or images.
3. **Forensic Science**: Tracking leaks of sensitive documents or media.
4. **Education**: Authenticating e-learning materials and certificates.

Performance Evaluation Metrics

We can use following matrices for evaluating the performance effectiveness of the algorithm used to embed the information into an image/video signal.

1. **PSNR (Peak Signal-to-Noise Ratio)**:
    - Measures the perceptual quality of the watermarked video.

$$\mathbf{PSNR} = \mathbf{10} \times \mathbf{log}\frac{\mathbf{255^2}}{\mathbf{MSE}}$$

2. **MSE (Mean Square Error)**:
    - Quantifies the variance between watermarked and the original signal.

$$\mathbf{MSE} = \frac{\mathbf{1}}{\mathbf{M} \times \mathbf{N}} \sum_{\mathbf{x=1}}^{\mathbf{M}} \sum_{\mathbf{y=1}}^{\mathbf{N}} \{(\mathbf{f}(\mathbf{x},\mathbf{y}) - \mathbf{f}'(\mathbf{x},\mathbf{y}))\text{^}\mathbf{2}\}$$

Here

- MSE – Mean Square Error
- PSNR – Peak Signal to Noise Ratio
- f(x,y) – Original Frame of the video

- f'(x,y) – Watermarked Frame of the Video.

The term "Peak Signal-to-Noise Ratio" (PSNR) is generally utilized for quantifying the similarity between signals: one of which is original and the other is modified version of the same signal. PSNR is defined using the Mean Square Error (MSE), which provides insight into the differences between the original signal and its modified version. PSNR is expressed in a logarithmic scale, offering a standardized measure of signal fidelity. In contrast, MSE is measured in a straightforward numerical scale. After extracting the watermark at the receiver's end, we assess its robustness by measuring the correlation between the recovered watermark and the original watermark. This correlation test helps evaluate how well the watermark survived various signal distortions or attacks, providing a measure of the watermarking system's resilience to tampering and ensuring reliable detection of the embedded information.
Digital watermarking is a powerful tool to address the challenges of copyright protection, authentication, and forensic tracking in video content. With continuous advancements in hybrid techniques, adaptive algorithms, and robust evaluation metrics, watermarking systems are becoming increasingly reliable. By achieving an optimal balance between robustness, imperceptibility, and payload capacity, digital watermarking offers a promising solution for securing multimedia content in the digital age.

1.7 Introduction to Face Detection Algorithm

Face detection is a CV task that involves identifying and locating human faces in images or videos. The main goal is to detect the presence and position of faces in various settings, such as photos, video streams, or security footage. Unlike facial recognition, which identifies individuals, face detection simply locates faces within a given scene. Over the past many years, detecting face has emerged as a critical component in variety of applications such as surveillance, human-computer interaction, and image search engines. The challenge of face detection arises from the vast variability in facial appearances, including changes in lighting, pose, expression, and occlusion. This article delves into the key principles, algorithms, and techniques used in face detection systems.

1.7,1 Evolution of Face Detection Algorithms

Face detection can be traced back to the early 1990s when the first real-time face detection system was developed. However, the idea of recognizing human faces dates back centuries, with early studies in psychology and neurobiology laying the groundwork for modern algorithms. Initially, face detection algorithms were computationally expensive and limited by

the processing power of early computers. It wasn't until the 2000s, with advancements in machine learning and computer vision, that face detection became more practical and widely implemented.

1.7,2 Key Approaches to Face Detection

There are several key approaches to face detection, each of which has evolved to handle specific challenges in recognizing faces across diverse conditions. The most notable methods include classical machine learning techniques, deep learning-based methods, and hybrid approaches.

1. Haar Cascade Classifiers (Classical Approach)

The Haar Cascade classifier, introduced by Paul Viola and Michael Jones in 2001, is one of the most well-known and influential face detection techniques. This method uses a machine learning algorithm based on Haar-like features, which are simple rectangular features that encode differences in intensity between adjacent regions of an image. The key steps in the Haar Cascade method are as follows:

- **Feature Extraction**: The algorithm uses Haar-like features that capture patterns in pixel intensity variations. These features are then computed for different regions of the image.
- **Training a Classifier**: A classifier, typically a decision tree or an ensemble of classifiers (such as AdaBoost), is trained on positive and negative images to distinguish faces from non-faces.
- **Cascade of Classifiers**: The classifier is arranged in a cascade structure, where simpler classifiers are used at the first stages to quickly eliminate negative regions (non-faces) and more complex classifiers are applied later to verify potential face regions.

This method was revolutionary for its time because of its speed and efficiency, enabling real-time face detection on standard computing hardware. However, it has limitations, such as sensitivity to changes in lighting, scale, and rotation.

2. Histogram of Oriented Gradients (HOG)

The HOG method, commonly used in conjunction with a Support Vector Machine (SVM), is another popular approach for object detection, including face detection. HOG extracts gradient-based features, which capture edge information, from images. The presence of the faces can then be classified using these features.

The main steps in the HOG-based face detection method are:

- **Gradient Computation**: The gradients of the image are computed to capture edge information, which is essential for detecting object shapes, such as the contours of a face.
- **Feature Representation**: The image is divided into small blocks, and the gradient histograms of each block are aggregated to form a feature vector.
- **Classification**: An SVM is trained to differentiate between faces and non-faces based on the extracted HOG features.

HOG-based methods have shown good performance in terms of accuracy, especially when used with a well-trained SVM. However, they are computationally more expensive than Haar-based methods and may struggle with large variations in pose or occlusions.

3. Deep Learning-Based Approaches

With the advent of deep learning, particularly Convolutional Neural Networks (CNNs), face detection has undergone significant advancements. CNNs are capable of automatically learning hierarchical features directly from raw pixel data, which allows them to achieve much better performance than classical methods. Some of the most successful deep learning-based face detection models include:

- **Single Shot Multibox Detector (SSD)**: This model detects faces in a single forward pass through the network by generating multiple bounding boxes at different scales and aspect ratios. SSDs are fast and suitable for real-time face detection.
- **You Only Look Once (YOLO)**: YOLO is a deep learning-based object detection algorithm that performs both classification and localization in a single pass. YOLO has been widely adopted for various applications, including face detection, due to its speed and accuracy.
- **Faster R-CNN**: This is an extension of the original CNN-based architecture that integrates a region proposal network (RPN) for generating potential bounding boxes in the image. Faster R-CNN improves both speed and accuracy over previous methods.

Deep learning-based methods, particularly CNNs, have significantly outperformed traditional techniques in terms of accuracy, robustness to pose variations, and performance under real-time conditions. These models, however, require large datasets and significant computational resources for both training and deployment.

4. Region-based Approaches

Another important category of face detection methods involves region-based algorithms, such as the Region-based Convolutional Neural Networks (R-CNN) and its variants (Fast R-CNN,

Faster R-CNN). These methods focus on identifying potential face regions in an image and then applying a classifier to confirm the presence of faces.

- **R-CNN**: This method first generates region proposals using selective search, which identifies potential objects in the image. Each proposed region is then fed into a CNN for classification.
- **Fast R-CNN**: This improves upon R-CNN by using a single pass through the CNN for all proposed regions, making it more efficient.
- **Faster R-CNN**: Faster R-CNN integrates a region proposal network (RPN) directly into the model, eliminating the need for selective search and further improving detection speed.

Region-based methods are known for their high accuracy and robustness, but they can be slower than single-shot methods like YOLO and SSD.

1.7.3 Mathematical Explanation of Face Detection Algorithms

In order to understand how face detection algorithms work mathematically, it is essential to look at the fundamental methods that drive these systems. At the core, these methods rely on mathematical techniques such as image processing, machine learning, and optimization algorithms. Here, we'll provide a mathematical explanation for two key face detection techniques: **Haar Cascade Classifiers** and **Convolutional Neural Networks (CNNs)**.

1. Haar Cascade Classifiers:

The Haar Cascade method, introduced by Viola and Jones, is one of the earliest and most popular techniques in face detection. The method relies on **Haar-like features** and a **cascade of classifiers** to identify objects (in this case, faces) in an image.

Step 1: Haar-Like Features

The Haar-like features are simple rectangular features based on differences in pixel intensity. These features can be computed efficiently using integral images, which allow for fast calculation of pixel sums over rectangular regions.

Given an image II, the value of a Haar-like feature in a rectangular region of the image can be expressed as:

$$f = \sum_{(x,y)\in A} I(x,y) - \sum_{(x,y)\in B} I(x,y)$$

Where:

- I(x,y)I(x, y) is the intensity value at pixel location (x,y)(x, y).

- AA and BB are two rectangular regions within the image, where the sum of intensities in region AA is subtracted from the sum of intensities in region BB.

Step 2: Integral Image

The **integral image** is a key mathematical concept that allows Haar-like features to be computed efficiently. For an image II, the integral image IIII is defined such that each pixel value at (x,y)(x, y) contains the sum of all pixels above and to the left of that point, including the point itself:

$$II(x,y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i,j)$$

Using the integral image, the sum of pixel intensities in any rectangular region can be computed in constant time, which is crucial for real-time face detection.

Step 3: Cascade Classifier

The cascade classifier consists of several stages, each containing a simple classifier that tests for the presence of a face. Each stage performs a binary classification: face or non-face. The stages are designed in such a way that the classifier quickly eliminates negative regions (non-faces) in early stages, allowing the later stages to focus on harder cases (true faces).

The classifier in each stage is trained using **AdaBoost**, a machine learning algorithm that selects a weighted combination of weak classifiers (simple decision stumps). The weak classifiers hih_i make a decision at each stage, and AdaBoost combines these weak classifiers into a strong classifier HH:

$$H(x) = \sum_{i=1}^{T} \alpha_i h_i(x)$$

Where:

- TT is the total number of weak classifiers.
- αi\alpha_i is the weight of the ii-th weak classifier, determined by AdaBoost based on its accuracy.

2. Convolutional Neural Networks (CNNs):

Convolutional Neural Networks (CNNs) have become the dominant approach for face detection, particularly in deep learning-based methods. A CNN automatically learns features from raw pixel data, and the overall process involves several mathematical operations, such as convolution, pooling, and backpropagation.

Step 1: Convolutional Layer

The first key operation in a CNN is the **convolution** operation. The goal of this operation is to apply a set of filters (also called kernels) to the input image to extract features like edges, textures, and patterns. Mathematically, the convolution of an image I(x,y)I(x, y) with a filter FF at position (x,y)(x, y) is given by:

$$(I * F)(x, y) = \sum_{i,j} I(x + i, y + j)F(i, j)$$

Where:

- I(x,y)I(x, y) is the image at pixel (x,y)(x, y).
- F(i,j)F(i, j) is the filter or kernel, which slides over the image.
- ∗* denotes the convolution operation.

After convolution, the output is a feature map, which is passed through an **activation function**, typically the Rectified Linear Unit (ReLU), which is defined as:

$$\text{ReLU}(z) = \max(0, z)$$

This step introduces non-linearity into the network, allowing it to learn more complex patterns.

Step 2: Pooling Layer

To reduce the spatial dimensions of the feature maps while retaining the most important features, CNNs use **pooling** operations, such as **max pooling**. Max pooling takes a rectangular region of the feature map and selects the maximum value within that region. The operation is defined as:

$$\text{MaxPool}(x, y) = \max(I(x', y'))$$

Where I(x′,y′)I(x', y') represents the values in the region surrounding the (x,y)(x, y) pixel. Pooling helps reduce the computational complexity and makes the network more invariant to small translations in the input image.

Step 3: Fully Connected Layer

After several convolutional and pooling layers, the output is typically flattened into a 1D vector and passed through one or more **fully connected (dense) layers**. The fully connected layer uses a weighted sum of its inputs to compute an output:

$$z = \sum_i w_i x_i + b$$

Where:

- xix_i are the inputs from the previous layer.
- wiw_i are the weights learned during training.
- bb is the bias term.

The output of the fully connected layer is passed through a final activation function (often a softmax or sigmoid) to produce the final classification result.

Step 4: Loss Function and Backpropagation

The network learns to minimize the error between the predicted outputs and the true labels during training. The error is quantified using a **loss function**. For face detection, this is often a binary cross-entropy loss, defined as:

$$L(y, \hat{y}) = -\left(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})\right)$$

Where:

- y is the true label (1 for face, 0 for non-face).
- y^ is the predicted probability of the face class.

The loss is minimized using **gradient descent** and **backpropagation**, where the gradients of the loss with respect to the weights are computed and used to update the weights.

Mathematically, face detection relies on a combination of image processing techniques and machine learning methods. In the Haar Cascade method, simple image features and classifiers work together to locate faces, while CNNs use powerful convolutional and pooling layers to automatically extract features from raw image data and make predictions. The combination of these mathematical operations forms the backbone of modern face detection systems, enabling accurate and efficient face detection even in complex, real-world environments.

1.7.4 Challenges in Face Detection

Despite the progress in face detection algorithms, several challenges remain in real-world scenarios:

- **Variation in Pose**: Faces can appear at different angles, making it difficult for algorithms to detect faces when they are turned sideways or tilted.
- **Lighting Conditions**: Variations in lighting, such as shadows or overexposure, can negatively impact face detection accuracy.

- **Occlusion**: Faces may be partially blocked by objects like hands, hair, or other people, which can make detection more challenging.
- **Scale**: Detecting faces at different scales in images, from very close-ups to distant faces, requires algorithms to be multi-scale and adaptive.
- **Real-time Processing**: Real-time face detection, such as in video streams or surveillance systems, requires algorithms to be fast without compromising accuracy.
-

1.7.5 Applications of Face Detection

Face detection has broad applications across numerous fields, including:

- **Security and Surveillance**: Face detection is widely used in surveillance systems for monitoring public spaces and identifying individuals in crowds.
- **Biometric Authentication**: Face detection plays a crucial role in biometric systems for security purposes, such as unlocking smartphones or verifying identities.
- **Human-Computer Interaction**: Face detection is integral to systems that allow users to interact with devices based on facial expressions or gestures.
- **Social Media**: Many social media platforms use face detection for tagging people in images or videos.

Face detection has evolved from basic template-matching methods to sophisticated deep learning-based systems capable of handling the complexities of real-world environments. With continued advancements in machine learning and artificial intelligence, face detection is becoming more accurate, efficient, and robust. Although challenges remain, face detection algorithms have already proven their value in various industries, revolutionizing how we interact with technology and enhancing security systems globally. As computational resources continue to improve and new methods emerge, the future of face detection looks promising, with even greater accuracy and broader applications on the horizon.

1.8 Literature Review

Detailed Review

Digital video, composed of sequential frames containing both vital and redundant signals, has become integral to modern storage, transmission, and distribution across networks. As one of the most prevalent multimedia formats, video watermarking has gained prominence for

ensuring copyright protection in an era where creators are increasingly concerned about unauthorized use and distribution of their content.

Digital Watermarking: A Brief Overview

Digital watermarking is a sophisticated technique that embeds concealed information, such as a creator's name, company logo, or ownership mark, into a video or image. This hidden data, imperceptible to the casual observer, requires specific algorithms to decode, thereby providing proof of ownership and safeguarding intellectual property.

Domains in Video Watermarking

There are two primary domains for embedding watermarks in videos:

1. **Spatial Domain:** Known for its high perceptibility, spatial domain techniques preserve the original quality of the video. This approach can embed a watermark equal in size to the frame itself. However, its robustness against attacks is comparatively low.
2. **Transform Domain:** Offering superior robustness, transform domain techniques utilize mathematical transformations such as Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Singular Value Decomposition (SVD). This makes them more resilient against various distortions and attacks, making them the preferred choice for watermarking.

Evolution of Video Watermarking Techniques

The development of video watermarking began with spatial domain techniques and gradually advanced to transform domain methods, which further evolved to hybrid techniques combining multiple transforms. These innovations aimed to enhance robustness while maintaining perceptibility.

Key Contributions in Video Watermarking Research

Numerous researchers have contributed significantly to the field:

1. **Podilchuk & Wolfgang**: Explored human visual system properties for image and video watermarking using DCT, suggesting further research to improve robustness.
2. **Langelaar et al.**: Reviewed existing spatial and frequency domain techniques, highlighting their mathematical foundations and practical applications.
3. **Chandramouli et al.**: Differentiated between steganography and watermarking while outlining their applications and requirements.
4. **Paul**: Discussed spatial and frequency domain techniques and suggested future research combining multiple transforms.

5. **Kutter & Petitcolas**: Provided mathematical metrics for evaluating watermarking techniques, emphasizing robustness and perceptibility.

Advanced Techniques and Notable Studies

- **LSB Techniques**: Researchers such as Yahaya and Lee proposed Least Significant Bit methods for embedding watermarks, focusing on balancing perceptibility and security.
- **Transform-Based Approaches**: Innovations like those by Kamlakar and Santhi combined DWT and SVD to achieve robustness against noise, rotation, and compression attacks.
- **Hybrid Techniques**: Studies by Huang and Al-Khatib introduced hybrid methods leveraging DCT, DWT, and SVD, ensuring resilience against complex attacks.
- **Machine Learning Methods**: Studies from various authors shown the face detection algorithms using various methods.

1.9 Applications and Future Directions

Video watermarking techniques are applied in fields requiring copyright protection, such as media distribution, secure communications, and intellectual property management. Future research aims to enhance robustness against emerging attack methods while maintaining low computational overhead. Combining multiple domains and leveraging machine learning for adaptive watermarking are promising directions for further exploration.

By integrating advanced methods and hybrid approaches, the field of digital video watermarking continues to evolve, providing enhanced tools for protecting intellectual property and ensuring the authenticity of digital content.

Additional Review

Reference 40

- Focus: Identifies watermarks and evaluates their robustness using machine learning.
- Limitation: Does not explore machine learning and frequency domain approaches for dual security.

Reference 41

- Focus: Hybrid watermarking scheme using various frequency domain transformations to improve robustness.
- Limitation: Excludes machine learning and dual security aspects.

Reference 42

- Focus: Combines DWT, PCA, and SVD for embedding copyright watermarks in videos.
- Limitation: Machine learning methods are not addressed.

Reference 43

- Focus: Dual-domain watermarking for image authentication and restoration with robust performance.
- Limitation: Does not incorporate machine learning or address video watermarking.

Reference 44

- Focus: Examines deep learning for enhancing robustness and imperceptibility of watermarks.
- Advantage: Highlights frequency coefficient modifications.

Reference 45

- Focus: Medical image watermarking using Riesz wavelet and SVD with high imperceptibility and robustness.
- Limitation: Does not explore machine learning techniques or dual security.

Reference 46

- Focus: Compares spatial (LSB) and frequency domain (DWT) techniques, favouring DWT for copyright protection.
- Limitation: Machine learning is not addressed.

Reference 47

- Focus: Uses DWT, DCT, and genetic programming for secure watermarking of medical images.
- Limitation: Machine learning is not incorporated.

Reference 48

- Focus: Employs genetic algorithms for dual watermarking, enhancing security and resistance to manipulation.
- Limitation: Excludes video watermarking.
- Authors: Meng Li, Qi Zhong, Leo Yu Zhang

Reference 49

- Focus: Watermarking deep neural networks using frequency domain methods to safeguard against piracy.
- Limitation: Video watermarking is not addressed.

Reference 50

- Focus: A deep-learning-based video watermarking framework improving resistance to distortions.
- Limitation: Does not consider dual security using frequency domain approaches.

Reference 51

- Focus: Uses machine learning in wavelet transform domain for adaptive watermarking.
- Limitation: Video watermarking is not addressed.

Reference 52

- Focus: Combines DWT with machine learning techniques like SVM and PCA for robust watermarking.
- Advantage: Balances security and robustness effectively.

Reference 53

- Focus: Uses SVR in the DCT domain for imperceptible and robust image watermarking.
- Limitation: Video watermarking is not covered.

Reference 54

- Focus: Machine learning enhances watermarking through feature optimization.
- Advantage: Compares algorithms with DWT, KNN, and PNN for improved performance.

Reference 55

- Focus: Neural networks ensure robust and imperceptible watermarking for videos.
- Advantage: Effective against frame drop and processing attacks.

Reference 56

- Focus: Twofold encrypted watermarking combining SVD, Arnold encryption, and BCH codes for robustness.
- Limitation: Machine learning techniques are not included.

Reference 57

- Focus: Non-blind watermarking with Q-learning and matrix factorization for high PSNR values.
- Limitation: Primarily focuses on spatial and frequency domains.

Reference 58

- Focus: Examines ML model watermarking, its vulnerabilities, and protection strategies.
- Limitation: Does not explore image and video watermarking for dual security.

Reference 59

- Focus: Combines DWT and DCT for high imperceptibility and robustness.
- Limitation: Excludes machine learning or dual security in video watermarking.

Reference 60

- Focus: Cascading DWT and PCA functions for effective image watermarking using machine learning.
- Limitation: Does not address dual security for both image and video watermarking.

Existing Research

- Spatial domain techniques: Simple but prone to attacks.
- Transform domain methods: Improved robustness using DCT, DWT, and SVD.
- Hybrid approaches: Combine methods for optimal performance.
- Embedding watermarks in identified region of interest to improve the security of the embedded message.

1.10 Relationships between Constructs

- **Robustness vs. Imperceptibility**: Balancing visibility and durability.
- **Payload Capacity vs. Computational Complexity**: Addressing efficiency concerns.

Research Gaps

Existing methods often compromise robustness for imperceptibility or vice versa. Few studies explore hybrid methods with systematic performance evaluation. The watermark is often embedded in the entire part of the image/video and hence easy to identify that the watermarking process is carried out on the source.

Problem Statement

The lack of a comprehensive framework combining robustness, imperceptibility, and efficiency highlights the need for advanced hybrid watermarking techniques along with the region of interest where the watermarking is to be carried out.

Contribution

- Enhances watermarking performance using hybrid methods.
- Embedding the watermarks in the region of interest rather than on the entire source.
- Using the machine learning approach to identify the region of interest.

- Provides a scalable framework for real-world applications.
- Offers policy implications for digital rights management.

Objectives of the Research

- Develop hybrid digital watermarking methods for video.
- Evaluate robustness against common attacks.
- Optimize imperceptibility and computational efficiency.

Hypotheses / Research Questions

- **H1**: Hybrid watermarking methods improve robustness and imperceptibility over single-domain techniques.
- **H2**: Transform domain methods are more resilient to compression attacks than spatial domain methods.
- **RQ1**: What are the trade-offs between robustness and computational efficiency in hybrid methods?

1.11 Methodology

Research Design

- **Type**: Descriptive and exploratory.
- **Strategy**: Quantitative analysis of watermarking methods.
- **Framework**: Experimental comparison of spatial, transform, and hybrid techniques.

Tools Used

- MATLAB and Python for algorithm development and testing.
- PSNR and MSE for evaluating imperceptibility.
- Attack simulations (compression, noise) for robustness assessment.