



(RESEARCH ARTICLE)



Dynamically selected watermark insertion using machine learning

Soppari. Kavitha, T Sri Vinay *, G Sai Teja and P H Rohith

Department of Computer Science and Engineering, ACE Engineering College, Hyderabad, Telangana, India.

World Journal of Advanced Research and Reviews, 2023, 18(03), 577–584

Publication history: Received on 28 April 2023; revised on 10 June 2023; accepted on 12 June 2023

Article DOI: <https://doi.org/10.30574/wjarr.2023.18.3.1078>

Abstract

Watermarking is a process of embedding one image into another image, used to protect the image from copyright infringement; the main aim of this watermarking is to provide ownership assertion- only rightful owner can extract the watermark from image to prove the ownership. Day-to-day the attacks on digital content (like images) are increasing, but still the same traditional watermarking process is used for content protection. In traditional watermark embedding process static watermarks are used at the source which may lead to compromise of algorithm. To avoid this, in this paper a very new way of watermarking is introduced with the help of object detection, along with the concept of content-related watermarking by dynamically selecting the watermark image.

Keywords: Watermarking; Object Detection; Watermark; Digital Content.

1. Introduction

Along with technology, attacks on digital content like images are increasing as well. Watermarking is the one of the solution to protect the ownership of the digital property. Image watermarking can be termed as embedding a watermark image into a cover image. The image which is needed to be protected is known as cover image and an image which will be embedded into the cover image is known as watermark image. After the watermarking process, we can't see the watermark. [1] Generally there are few types of watermarking schemes like Spatial Domain, Crypto-enabled spatial domain, Transform Domain and Crypto-enabled transform domain etc. Each watermarking technique will have its own advantage and disadvantage. There are many object detection techniques like R-CNN(Region-Based Convolutional Neural Networks), FAST R-CNN, FASTER R-CNN, MASK R-CNN, YOLO, SSD, RETINANET, REFINEDET etc. In this paper a new technique which involves use of object detection and image watermarking is discussed. The main aim of this experiment is to dynamically select the watermark image i.e., when an user gives a cover image for watermarking, directly an image will not be embedded into the cover image instead an object will be detected from the given cover image and then the watermark which is related to the object detected from the cover image will be selected dynamically from a set of images then the watermark will be embedded into the cover image using watermarking technique. So, there is no need to select a watermark image by user.

2. Literature survey

However, many researchers proposed different watermarking techniques and object detection algorithms but in this paper we will combine the advantages of object detection and watermarking. We will discuss few related works.

A CNN based object detection method G-CNN is used, it can work without proposal algorithms. Object detection is implemented by using multi-scale grid of fixed bounding boxes. CNN is trained in such a way that it can move and scale a fixed multi-scale grid of bounding boxes towards objects. Any CNN network can be used as the backbone of the architecture. A spatial region of interest (ROI) is used where the feature of a box can be computed. From all possible

* Corresponding author: T Sri Vinay

bounding boxes an iterative search will be implemented by the G-CNN for object detection. Replacing the object proposals with fixed multi-scale grid of boxes is the goal of this experiment [2]. For detecting the objects in real-time images a new approach is proposed with the help of CNN and deep learning. To increase the detection precision single shot multi-box detector is used. However, it is not suitable for detecting tiny objects. A combination of faster R-CNN with convolutional features and SSMBD with multi-scale contents is used in the approach. The algorithm consists of two stages where stage one is feature maps extraction and other one is application of small convolutional filters for object detection. By matching the default boxes with the ground truth boxes a high class confidence score is achieved. To tackle the difference in aspect ratio separate filters with default boxes are used. Fall in precision which is one of the major problems in previous methods, is improved by using multi-scale feature map and default boxes [3]. A work on the watermarking on the digital images based on the DWT (Discrete Wavelet Transform) and SVD (Singular Value Decomposition) algorithms are used. In their work DWT algorithm is used to provide better robustness and visible transparency so the algorithm converts the host color image and watermarking image into YUV and SVD is utilized to Y component. In these host and watermark images are decomposed into 2-level DWT on a low-frequency band and another frequency band is separated into $m \times n$ block size. As the color image is used here as a host image, SVD have been applied on each RGB component of a block is to find the singular values of the watermarking. By these methods the embedded image is produced. And to get the extracted watermark image the SVD (Singular Value Decomposition) and IDWT (Inverse Discrete Wavelet Transform) algorithms are used [4].

3. Dynamically Selected Watermark Insertion

The proposed system consists of three modules/stages, each module will play an important role in dynamic selection of watermark. The three modules are (i) Object detection module (ii) Dynamic selection of watermark (iii) Watermarking module. The flow of the experiment can be observed in the figure 3.1.

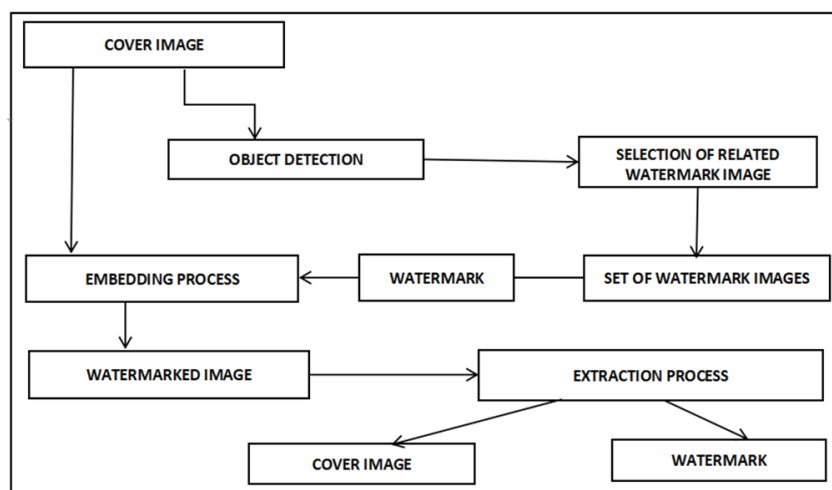


Figure 1 Architecture of Dynamically Selected Watermark Insertion using Machine Learning

When a cover image is given, an object from the image is detected with the help of object detection module, then the detected object name will be passed to the dynamic selection of watermark module. There will be few folders with the name of few objects, a random image (watermark) will be chosen from a folder which on the name of the object detected in the object detection module, then this watermark image is embedded into the cover image in the watermarking module. For the object detection module SSD Mobilenet is used and for watermarking Least Significant Bit (LSB) Technique is used.

3.1. Algorithm for Dynamically Selected Watermark Insertion

3.1.1. Step 1: Object Detection Module

The main reason for choosing the Mobilenet SSD is it offers a very good accuracy in object detection quickly compared to other architectures [5]. With MobileNet-SSD, you can address the model more quickly by using 8-bit integers rather than 32-bit floating. The model's input was a 300 by 300 pixel image, and its output addressed the position of the bounding box and the detection confidences (from 0 to 1) for each recognized object [6]. To determine whether the discovered object was legitimate, a detection confidence threshold of 0.5 was applied[7].

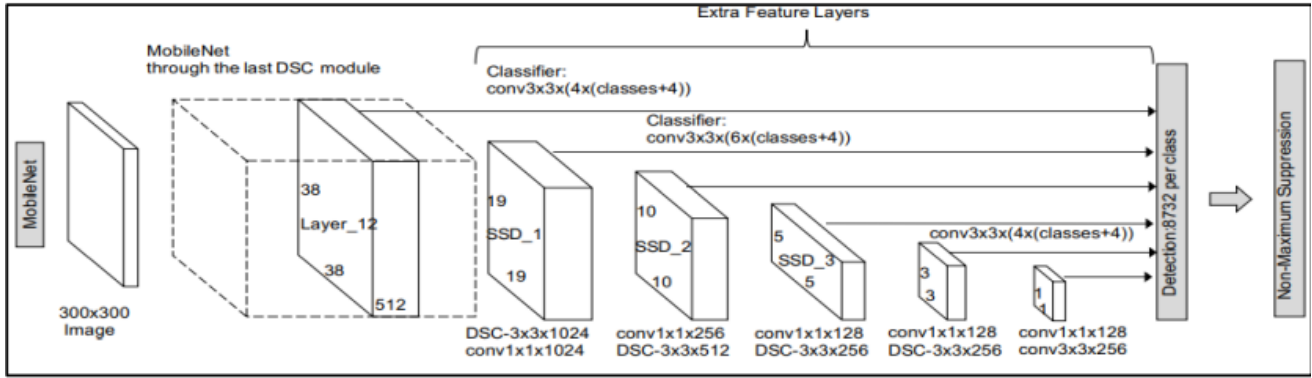


Figure 2 SSD network structure

The SSD (Single Shot MultiBox Detector) model is composed of several parts. The first part is the backbone network, which consists of a basic network and an additional feature extraction layer. The backbone network is responsible for extracting features from input images using deep neural networks. The second part is the original bounding box generation, where different default boxes are designed to extract feature maps at various scales. These default boxes serve as references for predicting the types and locations of objects in the images. The third part is the convolution prediction, which involves predicting the object categories and positions. The features within the default boxes are utilized to make these predictions.

Algorithm

The main process of the SSD algorithm can be summarized as follows:

- Input images are fed into the network, and features are extracted using the backbone network.
- Different default boxes are employed to extract feature maps at multiple scales.
- The features within the default boxes are utilized to predict the type and location of the target objects.
- The non-maximal suppression algorithm (NMS) is applied to select the most accurate prediction results that closely match the actual target boxes.

In the SSD model, a series of concentric default boxes are generated centered on the midpoint of each point on the feature map. The offset used for generating these default boxes is set to 0.5.

To make predictions, the model utilizes m feature maps of different sizes. The scale of the smallest underlying feature map is set to a value of 0.2, while the scale of the largest (top) feature map is set to a value of 0.95. The scales of the intermediate feature maps are calculated according to Equation (1).

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m - 1} (k - 1) \quad (1)$$

$$k \in [1, m]$$

To calculate the width (w) and height (h) of the default box using different ratio values (r_a), we can utilize equations (2) and (3). Here's a summary of the process:

- For each ratio value (r_a) in the set $\{1, 2, 3, 1/2, 1/3\}$, perform the following steps:
- Compute the width (w) and height (h) of the default box using equations (2) and (3):

$$w_k^a = s_k \sqrt{a_r} \quad (2)$$

$$h_k^a = s_k / \sqrt{a_r} \quad (3)$$

When ratios = 0, then according to (4) specific scale will be calculated

$$s_k^* = \sqrt{s_k s_{k+1}} \quad (4)$$

The SSD (Single Shot MultiBox Detector) algorithm uses a loss function that combines the confidence loss and the position loss based on the output of the prediction part. The formula for the loss function is given by equation (5):

$$L(z, c, l, g) = \frac{1}{N} (L_{conf}(z, c) + \alpha L_{loc}(z, l, g)) \quad (5)$$

Here, N represents the number of prediction boxes that match the ground truth object box. $L_{conf}(z, c)$ represents the confidence loss, and $L_{loc}(z, l, g)$ represents the position loss. The variable z indicates whether the prediction box matches the ground truth target box, with z = 1 indicating a match and z = 0 indicating no match. The variable c represents the confidence of the prediction box, and l represents the information on the location of the prediction box. The variable g denotes the location information of the ground truth object box.

The weight coefficient D determines the weight relationship between the confidence loss and position loss. In most cases, the two losses are given equal weight, and D is set to 1. In summary, the loss function in the SSD algorithm considers both the confidence loss and the position loss, with the aim of optimizing the model's ability to detect objects accurately and precisely [8].

3.1.2. Step 2: Dynamic Selection of Watermark

The second stage is dynamic selection of watermark, when the object is detected in the first stage (Object Detection) the object name will be passed to this stage. There will be few folders with the name of different objects, containing the watermark images related to that particular object. A watermark image will be randomly selected from the folder then it is passed to the third stage. For example there are few folders with the names car, bat, bicycle etc. There will be few images inside the folders. When a cover image is given, as discussed an object will be detected from the image. Assume that detected object is car, from folder named car a watermark image is randomly selected for watermarking.

$W_i = \text{Rand}(F_{obj})$

$W_i = \text{Watermark Image}$

$F_{obj} = \text{Folder with Object name}$

Rand = Function to randomly select an image.

Algorithm

- Selecting the folder with the name of the object detected.
- Choosing a random image as watermark for embedding
- Passing the cover and watermark for watermarking step.

3.1.3. Step 3: Watermarking Module

The Least Significant Bit (LSB) is used for watermarking, it is a basic technique used for inserting watermarks in spatial-domain watermarking. It involves directly manipulating the pixel values of an image by altering the least significant bit. This minor change in the pixel values allows for the application of a watermark. The process of inserting and extracting the watermark is straightforward and efficient. The LSB method produces watermarked images with high perceptual quality, making it commonly used in fragile watermarking techniques [9]. The process begins by selecting a host image

that will serve as the carrier for the watermark. Additionally, a secret image or watermark to be embedded is chosen. Both the host image and secret image are then converted into binary format, representing each pixel value as a sequence of binary digits. It is important to ensure that the size of the secret image is smaller than or equal to the size of the host image. The embedding process involves iterating through the pixels of the host image and replacing the LSB of each pixel's color values (red, green, and blue channels) with the corresponding bit from the secret image. This effectively hides the secret image within the host image. The process continues until all the pixels from the secret image have been embedded. Once completed, the modified host image, now containing the hidden secret image, is saved.

Algorithm for Embedding:

- Read the cover image and the watermark image.
- Check if the size of the watermark image is smaller than or equal to the cover image. If not, return an error.
- Convert the cover image and the watermark image into grayscale, if they are in color.
- Traverse each pixel of the cover image in raster-scan order.
- For each pixel, retrieve the corresponding pixel from the watermark image.
- Extract the least significant bit (LSB) from the cover image pixel and replace it with the LSB of the watermark image pixel.
- Update the modified pixel in the cover image.
- Repeat steps 5-7 for all pixels in the cover image.
- Return the watermarked image.







Algorithm for Extracting

- Read the watermarked image.
- Convert the watermarked image into grayscale, if it is in color.
- Create an empty image for the extracted watermark with the same size as the watermarked image.
- Traverse each pixel of the watermarked image in raster-scan order.
- For each pixel, extract the least significant bit (LSB) and store it as the corresponding pixel in the extracted watermark image.
- Repeat steps 4-5 for all pixels in the watermarked image.
- Return the extracted watermark image.

4. Experimental Results










Three different cover images were chosen for watermarking, we will see how the watermarking process will occur. First the object in the cover image will be detected, then from the folder with the name of object will be chosen, then a watermark image will be selected dynamically. At last the chosen watermark will be embedded into the cover image. The same watermark may or may not be chosen if the same cover image is chosen for watermarking process.

Table 1 Embedding Process

S.NO	Different Cover Images	Detected Object Name	Dynamically selected image	Watermarked Image
1		Car		
2		Boat		

3		Bicycle		
---	---	---------	--	---

Table 2 Extracting Process

S.NO	Different Watermarked Images	Extracted Cover image	Extracted Watermark
1			
2			
3			

4.1. Performance Evaluation

4.1.1. PSNR

Peak Signal-to-Noise Ratio (PSNR) is a widely used objective quality metric to assess the fidelity of reconstructed or compressed images or videos. It measures the ratio between the maximum possible signal power and the power of the distortion introduced during the compression or reconstruction process. Higher PSNR values indicate lower perceptual differences between the original and reconstructed images, implying better visual quality. PSNR is often employed in image and video processing applications to evaluate the effectiveness of compression algorithms and to compare different encoding techniques. It can be defined by using Mean Squared Error (MSE)

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2.$$

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_i^2}{MSE} \right)$$

MAX_i = Maximum pixel density for the image.

MAX_i = 255, is the result of representing pixels with 8 bits per sample.

MAX_i = Using linear pulse code modulation with B bits per sample, the sample representation is 2^B-1

M = no of rows of pixels of the images

i,j represents index of the row and column

n represents the number of pixels of the image.



Figure 3 Car Image

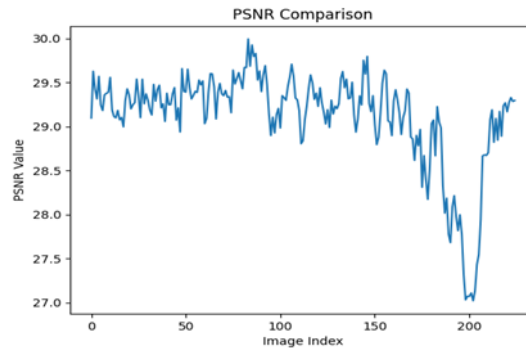


Figure 4 PSNR Plot for Car Image

4.1.2. SSIM

SSIM (Structural Similarity Index) is a perceptual quality metric that measures the similarity between two images. It takes into account luminance, contrast, and structural information to assess the perceptual difference. SSIM provides a score between 0 and 1, where higher values indicate greater similarity. It is commonly used in image processing tasks where preserving visual quality is crucial, such as compression and enhancement.

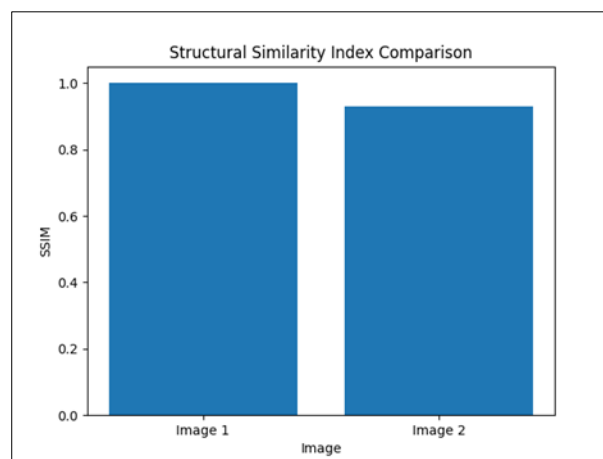


Figure 5 SSIM for Car Image

5. Conclusion

In this paper we used a concept of content related watermarking with the help of object detection, by dynamically selecting a watermark image and LSB for watermarking. This will make watermarking process very efficient and robust. The results showed a good quality of watermarked images and good PSNR and SSIM value are observed. Where traditional watermarking techniques are completely static in nature. There is further scope of improving the process of object detection and watermarking process.

Compliance with ethical standards

Acknowledgments

We would like to thank our guide Mrs. Soppari.Kavitha for her continuous support and guidance. Also, thankful to our project coordinator Mr.CH Vijay Kumar and we would like to express our gratitude to Dr. M.V.VIJAYA SARADHI, Head of the Department of Computer Science and Engineering, Ace Engineering College who was a continual source of inspiration.

Disclosure of conflict of interest

We have no conflicts of interest to disclose. All authors declare that they have no conflicts of interest.

References

- [1] "A Survey on crypto-enabled watermarking schemes" by Kavitha Soppari, Sri Vinay Tanniru, Sai Teja Gurjala, Rohith Pokala.
- [2] "G-CNN: an Iterative Grid Based Object Detector." by Mahyar Najibi, Mohammad Rastegari, Larry S.Davis.
- [3] "Object Detection System Based on Convolution Neural Networks Using Single Shot Multi-Box Detector" by Ashwani Kumari, Sonam Srivastava
- [4] "Robust Digital Watermarking for Digital Images based on DWT-SVD" by Jyoti Bala and Shweta Rai
- [5] "Shoe Detection Using SSD-MobileNet Architecture" by Ibai Gorordo Fernandez and Chikamune Wada
- [6] "Real Time Object Detection and Recognition using MobileNet-SSD with OpenCV" by Mr. Harshal Honmote, Mr. Shreyas Gadekar, Mr. Pranav Katta and Prof. Madhavi Kulkarni.
- [7] "Mobilenet-SSDv2: An Improved Object Detection Model for Embedded Systems" by Yu-Chen Chiu, Chi-Yi Tsai, Mind-Da Ruan, Guan-Yu Shen and TsuTian Lee
- [8] "Object detection system based on SSD algorithm" by Qianjun Shuai, Xingwen Wu
- [9] "Image Watermarking Scheme Using LSB and Image Gradient" by Zaid Bin Faheem, Mubashir Ali, Muhammad Ahsan Raza, Farrukh Arslan, Jehad Ali, Mehedi Masud and Mohammad Shorfuzzaman