

특집논문 (Special Paper)

방송공학회논문지 제25권 제2호, 2020년 3월 (JBE Vol. 25, No. 2, March 2020)

<https://doi.org/10.5909/JBE.2020.25.2.166>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

워터마크 및 해상도 적응적인 영상 워터마킹을 위한 딥 러닝 프레임워크

이 재 은^{a)}, 서 영 호^{a)}, 김 동 욱^{a)†}

Deep Learning Framework for Watermark-Adaptive and Resolution-Adaptive Image Watermarking

Jae-Eun Lee^{a)}, Young-Ho Seo^{a)}, and Dong-Wook Kim^{a)†}

요 약

최근 다양한 형태와 종류로 영상 콘텐츠를 가공하고 사용하는 응용분야가 급격히 증가하고 있다. 영상 콘텐츠는 고부가가치의 콘텐츠이므로 영상 콘텐츠의 제작 및 사용이 활성화되기 위해서는 이 콘텐츠의 지적재산권이 보호되어야 하며, 현재까지 그 방법으로 가장 널리 연구되고 있는 것이 디지털 워터마킹이다. 이에 본 논문에서는 딥 러닝 기반의 워터마크 삽입 및 추출 네트워크를 제안한다. 제안하는 방법은 호스트 영상의 비가시성(invisibility)을 보존하면서 악의적/비악의적 공격에 워터마크의 강인성(robustness)을 극대화하는 방법이다. 이 네트워크는 워터마크를 호스트 영상과 똑같은 해상도를 갖도록 변화시키는 전처리 네트워크, 변화된 호스트 영상과 워터마크 정보를 3차원적으로 정합하여 호스트 영상의 해상도를 유지하면서 워터마크 데이터를 삽입하는 네트워크, 그리고 해상도를 줄이며 워터마크를 추출하는 네트워크로 구성된다. 이 네트워크는 다양한 워터마크 영상과 다양한 해상도를 가진 호스트 영상에 대해 다양한 화소값 변경공격과 기하학적 공격을 실험하여 제안하는 방법의 비가시성과 강인성을 검증하고, 이 방법이 범용적이고 실용적임을 보인다.

Abstract

Recently, application fields for processing and using digital image contents in various forms and types are rapidly increasing. Since image content is high value-added content, the intellectual property rights of this content must be protected in order to activate the production and use of the digital image content. In this paper, we propose a deep learning based watermark embedding and extraction network. The proposed method is to maximize the robustness of the watermark against malicious/non-malicious attacks while preserving the invisibility of the host image. This network consists of a preprocessing network that changes the watermark to have the same resolution as the host image, a watermark embedding network that embeds watermark data while maintaining the resolution of the host image by three-dimensionally concatenating the changed host image and the watermark information, and a watermark extraction network that reduces the resolution and extracts watermarks. This network verifies the invisibility and robustness of the proposed method by experimenting with various pixel value change attacks and geometric attacks against various watermark data and host images with various resolutions, and shows that this method is universal and practical.

Keyword : convolutional neural network, deep learning, robust blind watermarking, invisibility, watermark-adaptive, resolution-adaptive

Copyright © 2020 Korean Institute of Broadcast and Media Engineers. All rights reserved.

“This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (<http://creativecommons.org/licenses/by-nc-nd/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered.”

1. 서론

최근 다양한 형태와 종류로 영상 콘텐츠를 가공하고 사용하는 응용분야가 급격히 증가함에 따라 영상 콘텐츠들의 무분별한 복제 및 유통으로 인한 문제가 발생하고 있다. 영상 콘텐츠는 고부가가치의 콘텐츠이므로 영상 콘텐츠의 제작 및 사용이 활성화되기 위해서는 이 콘텐츠의 지적재산권 보호가 매우 중요하며, 현재까지 그 방법으로 가장 널리 연구되고 있는 것이 디지털 워터마킹(digital watermarking)이다^[1]. 디지털 워터마킹 기술은 크게 호스트(host) 영상에 워터마크(watermark, WM) 영상을 삽입하는 삽입기와 워터마크를 삽입한 영상에 공격을 가한 영상에서 워터마크를 추출하는 추출기로 구성된다. 일반적으로 호스트 영상에 워터마크를 강하게 삽입할수록 삽입된 WM의 비가시성(invisibility)은 저하되고 공격에 대한 강인성(robustness)은 향상되는 상보(trade-off)관계를 갖는다.

기존에는 결정론적(deterministic) 알고리즘 기반의 워터마킹 방법이 많이 연구되어 왔으며, 비가시성을 위해 공간 영역에서 처리하는 연구보다 주파수 영역을 이용하는 연구가 많이 제안되었다^[2-8]. 그 중 DCT^[2], DWT^[3,4], DFT^[5]을 이용하여 주파수 영역을 구하는 방법, QIM^[6-8]을 이용하는 방법 등 변환에 대한 많은 연구들이 진행되었다. 최근에는 딥 러닝(deep learning) 기반의 워터마킹 방법에 대한 연구가 활기를 띠고 있으며, 알고리즘 기반의 워터마킹 방법보다 우수한 성능을 보이고 있다^[9-15]. 가장 먼저 워터마킹 시스템을 딥 러닝으로 접목한 [9]는 워터마킹된 영상을 생성할 때만 네트워크를 구성하며 사용한 정보를 WM 추출시 사용하는 non-blind 워터마킹 방법을 제안하였다. [10]은 1차원적인 WM를 3차원으로 확장시켜 호스트 영상에 전역

적으로 정합(concatenation)하여 인코더(encoder)에서 워터마킹된 영상을 생성한다. 공격에 대한 강인성을 위해 노이즈 층을 수행하고, 노이즈된 영상을 디코더(decoder)에 입력하여 WM를 추출한다. 인코더는 컨볼루션 층으로, 디코더는 컨볼루션 층, 공간적 평균 풀링, 그리고 전결합 층으로 구성된다. 또한, 호스트 영상과 워터마킹된 영상의 대립적(adversarial) 손실 함수를 사용하는 것이 특징이다. [11]은 호스트 영상을 DCT 변환 층을 사용하여 주파수 대역으로 변환시킨 후 WM 정보를 정합시키고, 이를 원형(circular) 컨볼루션을 수행함으로써 WM 정보를 전역으로 확산시킨다. 그 다음 inverse-DCT 변환 층을 사용하여 다시 공간 대역으로 변환하고 strength factor를 곱해 호스트 영상에 더함으로써 워터마킹된 영상을 생성한다. 공격 층을 수행한 워터마킹된 영상에 다시 DCT 변환 층을 사용하여 주파수 대역으로 변환시킨 후, WM를 추출한다. [12]는 호스트 영상을 인코더를 수행하여 특징으로 변환하고 변환한 특징의 모든 채널에 WM 정보를 더한다. 이를 다시 디코더를 수행하여 생성한 특징을 strength factor를 곱해 호스트 영상에 더함으로써 워터마킹된 영상을 생성한다. 그 다음, 강인성을 위해 공격 시뮬레이션을 수행하고 검출기를 수행함으로써 WM를 추출한다. 이후에, [13]은 프로베니우스 놈(Frobenius norm)을 사용하여 공격 시뮬레이션을 대체하는 방식을 제안하였고, [14]는 워터마킹된 영상을 생성하는 인코더와 WM 정보를 추출하는 디코더를 먼저 학습한 다음, 공격 층을 추가하여 디코더만을 재학습하는 방식을 제안하였다. 또한, [15]는 [10]의 네트워크 구조와 방식을 그대로 가져가면서 학습할 때마다 손실 함수의 값이 큰 공격을 구하여 학습시키며 공격에 대한 오버피팅을 방지하는 방법을 제안하였다. 그러나 대부분의 연구가 워터마크를 특정 데이터로 지정하여 사용자가 해당 기술을 사용할 때 사용자의 워터마크 정보로 재학습을 수행해야하는 문제점을 안고 있다^[9,10,12,14,15]. 또한, 네트워크에 전결합 층 등이 포함되어 특정 해상도의 영상만 적용 가능하여 매우 제한적인 기술^[9,10,12,14,15]이거나 다양한 해상도에 대한 효과를 보이지 않아 실용성이 확보되지 못하였다^[11,13].

이에 본 논문에서는 사용자가 추가적인 학습과정 없이 다양한 워터마크 데이터와 다양한 해상도의 호스트 영상에

a) 광운대학교 전자재료공학과(Department of Electronic Materials Engineering, Kwangwoon University)

‡ Corresponding Author : 김동욱(Dong-Wook Kim)
E-mail: dwkim@kw.ac.kr
Tel: +82-2-940-5167

ORCID: <https://orcid.org/0000-0002-4668-743X>

※ 이 논문의 연구결과 중 일부는 한국방송·미디어공학회 “2019년 추계학술대회”에서 발표한 바 있음.

※ This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019R1F1A1054552).

· Manuscript received December 26, 2019; Revised January 30, 2020; Accepted January 30, 2020.

적용 가능한 딥 러닝 기반의 디지털 영상 워터마킹 프레임워크를 제안한다. 이 방법은 전처리 네트워크, WM 삽입 네트워크, 공격 시뮬레이션, WM 추출 네트워크로 구성되며, 모든 네트워크는 컨볼루션 층만을 사용하며, 호스트 영상이나 워터마크의 해상도와 관련된 계층을 사용하지 않는다. 전처리 네트워크는 워터마크 정보를 호스트 영상의 크기로 해상도를 증가시키고, 이를 호스트 영상과 정합하여 WM 삽입 네트워크에서 워터마킹된 영상을 생성한다. 공격에 대한 강인성을 향상시키기 위해 다양한 악의적/비악의적 공격에 대해 공격 시뮬레이션을 수행하고, 그 결과 영상을 WM 추출 네트워크에 입력하여 워터마크 정보를 추출한다. 이 네트워크를 학습할 때 워터마크 정보를 무작위로 생성하여 임의의 워터마크 영상에 대해 적용 가능하게 한다. 이를 다양한 해상도의 호스트 영상을 대상으로 다양한 공격을 실험하여 충분한 강인성을 갖고 있음을 보인다. 또한, 여러 가지의 워터마크 영상에 대한 실험 결과를 보이며 범용성과 실용성을 증명한다.

본 논문의 구성은 다음과 같다. 먼저, sII장에서 제안하는 프레임워크를 소개하고, III장에서 실험 과정을 설명하며 다양한 해상도를 가진 영상을 대상으로 다양한 공격에 대한 실험결과를 설명한다. 또한 이 결과를 최근에 발표된 방법들과 비교 및 분석하며 IV장에서 결론을 맺는다.

II. 제안하는 딥 러닝 기반의 워터마킹 프레임워크

그림 1에 제안하는 딥 러닝 프레임워크를 개략적으로 나타내었는데, ① 전처리 네트워크(호스트 영상, WM), ② WM 삽입 네트워크, ③ 공격 시뮬레이션, 그리고 ④ WM 추출 네트워크로 구성된다. 각 네트워크의 구조는 표 1에 네트워크 별로 보였다. 여기서 하나의 컨볼루션 계층을 컨볼루션 블록(Convolution Block, CB)이라 칭하며, 한 CB에는 컨볼루션, 배치정규화(Batch Normalization, BN), 그리고 활성화(Activation) 함수를 포함한다. 모든 블록에서 3×3 컨볼루션을 사용하므로, 각 블록의 IFM(Input Feature Map)은 좌, 우, 아래, 위로 각각 1 화소씩 0-패딩(padding)

된다. 각 네트워크의 마지막 CB는 배치정규화를 수행하지 않았으며, 활성화 함수로 삽입 네트워크와 추출 네트워크의 마지막 CB에는 tanh를, 나머지 CB는 ReLU(rectified linear unit)를 사용한다.

1. 전처리 네트워크

먼저, 전처리 네트워크는 호스트 영상(I_{host})과 워터마크 영상 각각을 따로 처리한다. 호스트 영상은 흑백영상(1 채널)을 사용하는데, RGB영상의 경우 YCbCr로 변환하여 Y영상을 사용한다. 호스트 영상은 한 개의 CB를 거쳐 워터마크를 삽입하기 적합한 특징(feature)을 추출한다.

워터마크 데이터(WM_{org})는 2차원 2진(binary) 영상을 사용한다. 4개의 CB와 2×2 평균 풀링(average pooling)을 간격=1로 수행한다. 이를 거쳐 호스트 영상과 같은 해상도를 가지면서 삽입에 적합한 특징들로 변환한다. 즉, 호스트 영상은 해상도를 그대로 유지하지만 워터마크 영상은 호스트 영상의 해상도에 맞추기 위해 CB에서 간격(stride) 조절을 통해 업샘플링(upsampling)된다. 그 결과의 워터마크 데이터는 강도조절을 위해 strength factor($s \in real$)를 곱해(그림 1의 ⊗) CB를 거친 호스트 데이터에 한 채널로 정합(concatenation)된다.

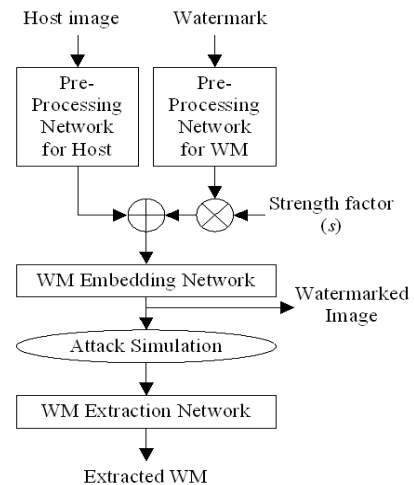


그림 1. 제안하는 워터마킹 네트워크 프레임워크
Fig. 1. The proposed watermarking network framework

표 1. 그림 1의 각 네트워크 구조
Table 1. Structure of each network in Fig. 1

Network	Kernel size	# of kernels	Stride	BN	Activation	Average-pooling (stride=1)
Pre-processing Network (Host)	3×3	64	1	×	-	×
Pre-processing Network (WM)	3×3	512	1/2	○	ReLU	○
	3×3	256	1/2	○	ReLU	○
	3×3	128	1/2	○	ReLU	○
	3×3	1	1/2	×	-	○
WM Embedding Network	3×3	64	1	○	ReLU	×
	3×3	64	1	○	ReLU	×
	3×3	64	1	○	ReLU	×
	3×3	64	1	○	ReLU	×
	3×3	1	1	×	tanh	×
WM Extraction Network	3×3	128	2	○	ReLU	×
	3×3	256	2	○	ReLU	×
	3×3	512	2	○	ReLU	×
	3×3	1	2	×	tanh	×

2. WM 삽입 네트워크

전처리 네트워크를 수행하여 정합된 호스트 데이터와 워터마크 데이터는 5개의 CB를 거쳐 워터마크가 삽입된 영상(I_{WMed})이 된다.

본 논문의 목표가 비가시성 워터마킹이기 때문에 워터마크가 삽입된 영상과 원 호스트 영상 간의 오차가 최소화되어야 한다. 이를 위해 이 두 영상 간의 MSE(Mean Square Error)를 전처리 네트워크와 삽입 네트워크의 손실함수(L_1)로 사용한다. 이를 식 (1)에 나타내었다.

$$L_1 = \frac{1}{MN} \sum_{i,j} [I_{host}(i,j) - I_{WMed}(i,j)]^2 \quad (1)$$

여기서 $M \times N$ 은 호스트 영상의 해상도이다.

3. 공격 시뮬레이션

공격에 대한 강인성을 확보하기 위해서는 추출 네트워크 뿐만 아니라 전처리 네트워크와 삽입 네트워크에서도 공격에 강인한 호스트 영상의 특징과 워터마크 특징을 추출하여 호스트 영상의 적절한 위치에 삽입하여야 한다. 이를 위해서는 네트워크 내에 공격을 위치시켜 학습이 이루어져야 하며, 따라서 본 논문에서는 네트워크 내에서 미분 가능한

공격 시뮬레이션을 수행한다. 공격은 악의적/비악의적 공격을 고려하여 화소값 변경공격(Pixel-value change attack)과 기하학적 공격(Geometric attack)을 수행한다. 공격의 종류(Attack), 공격 강도(Strength) 및 학습비율(Ratio)을 표 2에 나타내었는데^[10,11], 화소값 변경 공격은 7가지, 기하학적 공격은 3가지로 구성된다. 학습할 때 한 미니배치(mini-batch) 안에 공격을 가하지 않은 영상(Identity)과 함께 각 공격을 수행한 영상을 모두 포함시키며 미니배치마다 동일한 공격분포를 갖도록 구성한다. 미니배치 내에서의 공격 강도 분포도 거의 동일한 구성을 갖도록 한다.

표 2. 공격 시뮬레이션에 사용한 공격들
Table 2. Attacks used in the attack simulation

Attack type	Attack	Strength	Ratio
No attack	Identity	-	1/12
Pixel-value change attack	Gaussian filtering	3×3, 5×5, 7×7, 9×9	2/12
	Average filtering	3×3, 5×5	2/12
	Median filtering	3×3, 5×5	1/12
	Salt & Pepper	p=0.1	0.5/12
	Gaussian noise	sigma=0.1	0.5/12
	Sharpening	Laplacian 5-point stencil, 9-point stencil	1/12
	JPEG	Quality factor=50	1/12
Geometric attack	Rotation	(0~90°, random)	1/12
	Crop	(0.5~0.8, random)	1/12
	Dropout	(0.3, 0.9, random)	1/12

4. WM 추출 네트워크

필요시 삽입한 워터마크 데이터를 추출하기 위한 WM 추출 네트워크는 표 1과 같이 4개의 CB로 구성된다. 이 네트워크의 입력은 워터마크가 삽입된 영상에 공격을 가한 영상이다. 추출한 워터마크와 입력으로 사용된 원본 워터마크 영상의 MAE(Mean Absolute Error)를 추출 네트워크의 손실함수(L_2)로 사용한다.

$$L_2 = \frac{1}{XY} \sum_{i,j} |WM_o - WM_{ext}| \quad (2)$$

여기서 $X \times Y$ 는 워터마크 데이터의 해상도이다. 이 손실함수에 공격 시뮬레이션에 대한 손실함수가 포함된다.

5. 네트워크의 최종 손실함수

삽입한 WM 정보의 비가시성과 공격에 대한 강인성은 상보적인(trade-off) 관계를 갖는다. 따라서 비가시성과 강인성을 모두 만족시키기 위해 WM 삽입 네트워크(L_{emb})와 WM 추출 네트워크의 손실함수(L_{ext})를 식 (3)과 식 (4)와 같이 구성한다.

$$L_{emb} = \lambda_1 L_1 + \lambda_2 L_2 \quad (3)$$

$$L_{ext} = \lambda_3 L_2 \quad (4)$$

이 두 식에서 λ_1 , λ_2 , λ_3 는 하이퍼 파라미터(hyper-parameter)로 비가시성과 강인성을 조절하는 파라미터이다. λ_1 은 L_1 손실을 삽입 네트워크에 적용하는 강도, λ_2 는 L_2 손실을 삽입 네트워크에 사용하는 강도, λ_3 는 L_2 손실을 추출 네트워크에 사용하는 강도를 각각 나타낸다.

그림 1과 표 1의 네트워크 구성을 보면 제안하는 네트워크는 호스트 영상이나 각 CB의 IFM의 해상도를 정의하지 않으며, 어느 네트워크에도 전결합 층(fully connected layer)과 같은 해상도에 종속되는 층을 사용하지 않는다. 따라서 제안한 네트워크는 호스트 영상과 워터마크 데이터의 해상도에 무관하게 적용할 수 있다. 또한 워터마크된 영상은 저장 또는 배포되는 대상 데이터이므로 8-비트의 정밀도를 갖는 정수형 데이터로 변환하여 사용하고, 최종 추출된 워터마크 정보는 1-비트의 정밀도(2진수)의 정수형으로 추출되도록 한다.

III. 실험 및 결과

제안하는 네트워크의 비가시성 및 강인성을 평가하기 위해 다양한 실험을 수행하였다. 먼저 실험 환경을 설명하고, 다양한 공격과 해상도에 대한 실험을 수행하여 그 결과와 최신 연구결과와 비교 및 분석한다.

1. 구현 및 실험

1.1 실험 환경

본 연구의 구현 환경은 파이썬(Python)과 텐서플로우

(Tensorflow)이고, 사용한 PC는 Intel(R) Core(TM) i7-9700 CPU @3.00GHz, 64GB RAM을 갖고 있으며, 운영체제는 64-bit Windows, 그리고 GPU는 RTX 2080ti를 사용하였다. 이 PC로 네트워크를 훈련하는 데에는 최소 3일에서 최대 6일 정도의 시간이 소요되었다. 학습에서 미니배치(mini-batch)는 100으로 하였고, 최적화방법(optimizer)은 Adam을 하이퍼 파라미터 $\beta_1=0.5$, $\beta_2=0.999$, $\eta_1=0.0001$, $\eta_2=0.00001$ 으로 설정하였으며, 학습은 4000 에폭(epoch)까지 수행하였다.

1.2 데이터 셋

호스트 영상의 학습 데이터로 그레이(gray) 스케일 영상이 10,000장으로 구성된 BOSS 데이터 셋^[16]을 128×128 해상도로 스케일링(scaling)하여 사용하였고, 평가(test) 데이터로 그레이(gray) 스케일 영상이 49장으로 구성된 표준 시험 데이터셋^[17]을 128×128 해상도로 스케일링하여 사용하였다. 학습할 때 그림 1의 워터마크 데이터의 강도(s)는 1로 고정하였다. 워터마크 영상은, 학습할 때마다 8×8 해상도의 이진(binary) 스케일 영상을 무작위로 생성하여 사용하였고, 평가할 때는 다양한 실험에 대한 효과만을 확인하기 위하여 무작위로 생성한 영상으로 데이터셋을 구축하여 평가하였다.

워터마크 데이터에 제한이 없다는 것은 특정 워터마크 데이터에 맞춰서 학습된 것이 아니라는 것을 의미한다. 만약 특정 워터마크 데이터를 사용하여 학습한다면 그 워터마크 데이터에만 적용되어 다른 워터마크 데이터에는 성능이 떨어질 가능성이 높다. 즉, 사용자가 자신의 워터마크를 사용하기 위해서는 그 워터마크 데이터로 학습을 다시 수행하여야 한다. 따라서 제안한 네트워크가 임의에 워터마크 데이터에 적용 가능하기 때문에 사용자가 제안한 네트워크를 사용할 때 본인의 워터마크로 재학습할 필요가 없어 매우 실용적이고 범용적인 워터마크-적응적 워터마킹 방법이다.

한편, 네트워크의 결과 워터마킹된 영상과의 추출된 워터마크 모두 각 화소가 [-1, 1]이기 때문에 모든 호스트 영상과 워터마크 데이터를 [-1, 1]로 정규화를 수행한 후에 입력으로 사용한다. 단, 워터마크된 영상은 [0, 255]로 변환하여 출력하며, 변환된 데이터에 공격 시뮬레이션을 수행하고, 그 결과를 다시 [-1, 1]로 정규화하여 추출 네트워크에 입력한다.

1.3 성능 평가 방법

워터마킹 방법의 성능을 평가할 때 비가시성, 강인성, 그리고 수용력의 3가지를 주로 평가한다. 비가시성은 워터마크가 삽입된 영상과 원본 호스트 영상의 PSNR(Peak Signal Noise Ratio)로, 강인성은 추출한 워터마크 영상과 원본 워터마크 영상의 BER(Bit Error Ratio)로, 수용력은 워터마크 영상 해상도에 호스트 영상 해상도를 나눠서 측정한다. 본 논문에서는 수용력을 0.0039(8×8/128×128)로 고정하였다.

2. 실험결과

2.1 삽입한 워터마크의 비가시성

학습결과 공격을 가하지 않은 상태에서 워터마크가 삽입된 영상의 PSNR은 $s = 1$ 인 상태에서 평균 43.23[dB]이었으며, 이 때의 가중치 세트(weight set)를 테스트 셋에 적용한 테스트 결과는 40.58dB이었다. 그림 2에 호스트 영상(a)과 그 영상에 워터마크를 삽입한 영상 두 쌍의 예를 보이고 있는데, 육안으로는 두 영상의 차이를 거의 발견할 수 없을 정도로 높은 비가시성을 보이고 있다.



그림 2. 워터마크의 비가시성 결과의 예: (a) 호스트 영상, (b) 워터마킹된 영상
 Fig. 2. Examples of the watermark invisibility results: (a) host image, (b) watermarked image

2.2 다양한 공격에 대한 강인성 실험 결과

위에서 언급한 학습결과의 가중치 세트를 사용하여 테스트 셋의 영상을 대상으로 다양한 종류와 강도에 대한 워터마크 강인성 실험을 수행하였다. 먼저 공격을 가한 영상의 예를 그림 3에 보이고 있는데, 이 그림에서 보이는 대부분의 공격은 그 강도가 원영상(a)을 심하게 훼손할 정도여서 공격된 영상을 재사용하기는 어려운 수준이다. 따라서 이

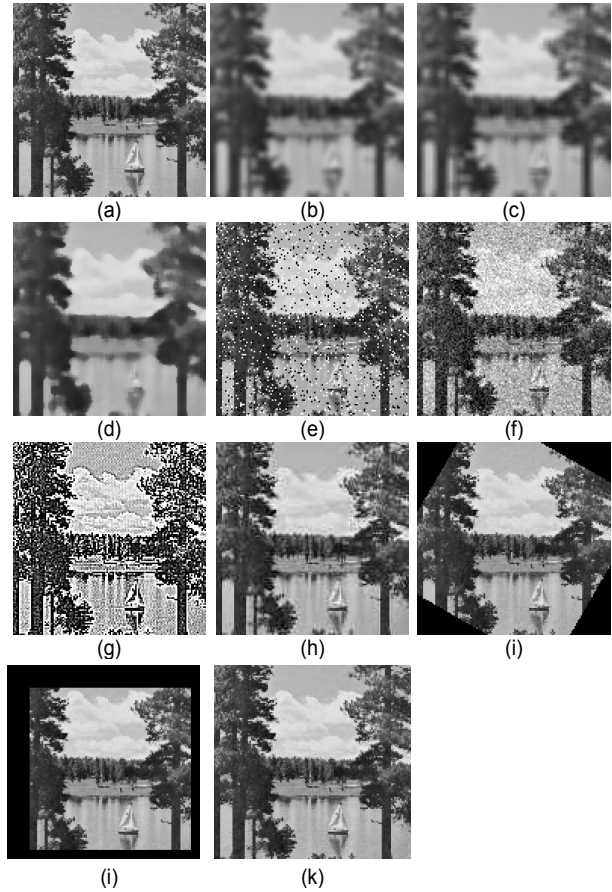




그림 3. 학습할 때 사용한 공격한 영상들: (a) 호스트 영상, (b) 가우시안 필터링 (7×7), (c) 평균 필터링 (5×5), (d) 중간값 필터링 (5×5), (e) Salt and Pepper 잡음첨가 (p=0.1), (f) Gaussian 잡음첨가 (sigma=0.1), (g) Laplacian 샤프닝(5-point stencil), (h) JPEG 압축(quality factor=50), (i) 회전 (30°), (j) Crop (p=0.5), (k) Dropout (p=0.5)

Fig. 3. Attacked images used in training (a) Host image (b) Gaussian Filtering (3×3), (c) Average Filtering (3×3), (d) Median Filtering (3×3), (e) Salt and Pepper noise addition (p=0.1), (f) Gaussian noise addition (sigma=0.1), (g) Laplacian sharpening(5-point stencil), (h) JPEG compression (quality factor=50), (i) Rotation (30°), (j) Crop (p=0.5), (k) Dropout (p=0.5)

런 강도의 공격은 워터마킹의 목적에 부합하지 않지만, 본 논문에서는 이런 공격과 그 이상의 공격강도도 고려한다.

표 3에 공격에 대한 강인성 실험결과를 보이고 있는데,

표 3. 공격 강인성 실험 결과 추출한 워터마크의 평균 BER 값
Table 3. Average BER values of extracted watermark resulting from the robustness experiments

Attack type	Attack	Strength	BER (%)		
			WM1 Random (average)	WM2 	WM3 
Pixel-value change attacks	No attack	-	0.7015	0.6696	0.6696
	Gaussian filtering	3×3	1.5944	1.7538	2.0089
		5×5	7.5255	7.3023	8.4503
		7×7	11.5115	11.2883	11.5115
		9×9	18.463	19.1964	17.5064
	Average filtering	3×3	4.273	3.9541	4.1135
		5×5	5.2296	5.3571	4.7832
	Median filtering	3×3	8.4184	7.8763	7.8763
		5×5	10.5548	10.8418	11.0969
	Salt and Pepper noise addition	0.01	0.861	0.7972	0.7972
		0.03	1.1798	1.0204	1.1161
		0.05	1.4031	1.2436	1.3393
		0.07	1.8176	1.5306	1.8495
	Gaussian noise addition	0.09	2.5829	3.1250	2.0408
		$\sigma=0.01$	0.8291	0.7972	0.7334
		$\sigma=0.03$	1.977	1.6582	1.977
		$\sigma=0.05$	6.0906	6.8878	5.6441
	Sharpening	5-point stencil	11.9898	12.8508	13.3291
		9-point stencil	3.2844	3.6671	3.5714
	JPEG	90	0.9566	0.8610	0.7653
70		4.2411	3.9860	4.2411	
50		8.0676	7.9082	9.0561	
30		14.8916	15.1148	14.8278	
10		31.4732	31.8240	33.4184	
Geometric attacks	Rotation	15	2.0727	1.8814	1.9133
		30	4.9107	4.8151	5.2296
		45	5.0383	5.1339	5.7398
		60	3.8265	3.6671	4.7194
		75	1.7857	1.8176	1.9452
	Crop	0.9	0.7015	1.1798	0.9247
		0.7	2.1365	4.3367	13.4566
		0.5	14.6365	16.7411	11.4796
		0.3	24.9681	21.3967	29.1773
	Cropout	0.1	39.6365	48.5013	38.361
		0.1	2.2003	3.0293	1.7538
		0.3	9.0561	12.4362	9.088
		0.5	17.0281	24.2666	20.9184
		0.7	24.0434	33.4184	25.4783
	Dropout	0.9	34.8533	44.9298	37.3724
		0.9	0.9247	0.9247	1.0523
		0.7	2.4554	2.2003	2.3278
0.5		6.25	5.4528	5.5166	
0.3		14.8916	15.5612	15.1148	
		0.1	34.1199	37.3087	34.7577

각 공격의 종류에 대해 대표적인 공격강도에 대한 실험결과와 BER 값들을 보이고 있으며, 이 값들은 테스트 셋의 영상들에 대한 평균값이다. 공격을 가하지 않은 워터마킹된 영상에서의 워터마크 추출오차율은 약 0.7% 정도이었으며, 어느 공격에서도 강도가 높아짐에 따라 BER이 포화되거나 증가하는 오버 피팅(over-fitting) 현상이 발생하지 않았다. 표 3(WM1)에서 보듯이 아주 강한 저역통과필터링(low-pass filtering)과 고압축의 JPEG 압축 공격을 제외한 화소값 변경 공격에는 높은 강인성을 보였다. 반면 회전을 제외한 기하학적 공격에서 강도가 높아짐에 따라 강인성이 많이 떨어졌다. 그러나 강도가 높은 화소값 변경 공격이나 기하학적 공격은 공격된 영상의 왜곡이 심하여 그 자체로서는 사용가치가 많이 떨어지기 때문에 제안한 방법의 효용성은 매우 높다고 사료된다.

2.3 다양한 워터마크에 실험 결과

본 연구의 목적 중 하나가 사용자가 임의의 워터마크 정보를 사용할 수 있도록 하는 것이다. 이를 위하여 추가 학습 없이 기 학습된 가중치들을 사용하여 여러 종류의 인위적으로 생성된 워터마크를 삽입하고 추출하는 실험을 진행하였으며, 그 결과 중 두 셋을 표 3에 WM2와 WM3 열에 사용한 워터마크와 함께 보였다. 표에서 보는 바와 같이 다소의 차이는 있지만 세 경우 모두 비슷한 BER값들을 보이고 있어 제안한 방법이 임의의 워터마크 정보를 사용할 수 있음을 확인하였다.

2.4 호스트 영상의 해상도에 따른 실험 결과

본 연구에서 제안하는 네트워크에서는 호스트 영상이나 워터마크 데이터의 크기(해상도)에 따라 변화하는 계층을 사용하지 않기 때문에 호스트 영상이나 워터마크 데이터의 크기와 무관하게 적용할 수 있다. 이에 호스트 영상의 해상도가 변화함에 따른 공격 강인성 실험을 수행하였는데, 고려한 호스트 영상의 해상도는 64×64부터 512×512까지였다. 이 실험에서는 해상도에 따른 공격 강인성의 변화를 확인하기 위하여 워터마크 수용률을 0.0039를 유지하였으며, 이에 따른 각 호스트 영상 해상도의 워터마크 크기를 표 4에 나열하였다. 워터마크에 대한 비가시성을 표 4의 마지막 열에 보이고 있는데, 해상도가 증가할수록 비가시성이 증가하였다.

표 4. 호스트 영상에 따른 워터마크 크기
 Table 4. Watermark size to the host image

Host image resolution	Watermark resolution	Invisibility [dB]
64×64	4×4	39.97
128×128	8×8	40.58
256×256	16×16	41.23
512×512	32×32	42.35

해상도 증가에 따른 강인성 실험 결과는 그림 4에 보이고 있다. 그림을 보면 화소값 변경 공격 중 Gaussian 잡음첨가 공격을 제외한 모든 공격에서 해상도가 증가함에 따라 공격 강인성이 향상되는 결과가 나타났고, Gaussian 잡음첨가, 회전, Dropout 공격에 대해서는 뚜렷한 경향성을 보이지 않았으며, 나머지 공격에 대해서는 오히려 강인성이 떨어지는 경향을 보였다. 그러나 해상도 증가에 따라 강인성이 떨어지는 정도가 크지 않아 비가시성을 대비한 결과를 감안하면 제안한 방법이 해상도가 증가하고 있는 최근 영상 추세에 매우 적합한 방법이라 사료된다.

3. 최근 연구와의 비교

제안한 방법의 성능을 확인하기 위해 최근 연구들과 제

안한 방법의 결과를 비교하였다. 최근 연구의 결과에서 수치를 제공한 방법이 유일하게 ReDMark^[11]이어서 먼저 이 방법과 비교한 결과를 표 4에 보였다. 여기서 제안한 방법의 비가시성은 워터마크 데이터의 강도 s 를 조절하여 PSNR을 ReDMark와 비슷하게 40.58[dB] 맞추어 실험하였다. 표에서 보는 바와 같이 Gaussian 잡음첨가 공격을 제외한 모든 공격에서 제안한 방법의 결과가 우수하였다.

또한 [14]에서 HiDDeN^[10]과 ReDMark^[11]를 [14]와 비교하였는데, 그 결과와 제안한 방법의 결과를 비교하여 표 5에 나타내었다. 이 비교에서 워터마크 비가시성을 맞추기 위해 공격 강도를 $s = 2.75$ 로 조절하였다. 그 결과, crop (0.035)를 제외한 모든 공격에서 [10]과 [11]보다 우수하였고, [14]에 비해서는 JPEG 공격만 더 좋은 결과를 보였다. 특히 제안한 방법은 JPEG 공격에서 굉장히 우수한 결과를 보였으나, crop ($p=0.035$) 공격에서 유독 안 좋은 결과를 보였다. 이 공격은 전체 영상의 3.5%만을 사용하는 매우 강한 공격으로, 워터마킹 분야에서는 무의미한 공격으로 간주된다. 또한, [14]의 경우 표 5의 공격만을 대상으로 학습을 수행하였기 때문에 이 공격에는 특히 강한 결과를 보이고 있으나, 그 외의 공격에 대해서는 데이터를 제시하지 않아 결과를 알 수 없다. crop(0.035) 공격을 제외한 모든

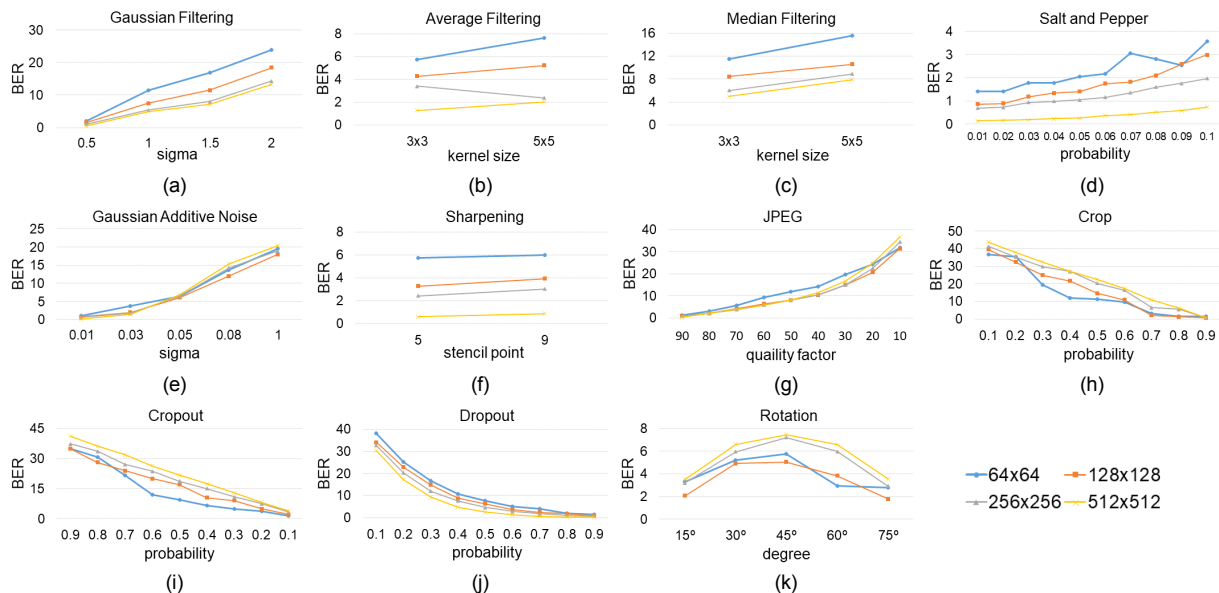


그림 4. 호스트 영상의 다양한 해상도에 대한 실험 결과로 추출한 워터마크의 BER 값
 Fig. 4. BER values resulting from the experiments for the various resolutions of host image

공격에서 제안한 방법이 전반적으로 좋은 성능을 보여 제안한 방법이 더욱 일반적인 효용성을 보인다고 판단된다.

표 5. ReDMark^[11]와의 비교
Table 5. Comparison with ReDMark^[11]

Attack	Strength	ReDMark	Proposed
PSNR		40.24 [dB]	40.58 [dB]
No attack	-	-	0.7015
Gaussian filtering	radius=1	8.6	7.1429
	radius=1.6	39	9.7258
	radius=2	-	12.7232
Median filtering	3×3	13.4	8.4184
	5×5	-	10.5548
Salt and pepper noise addition	0.02	2.9	1.0204
	0.6	4.5	1.5306
	0.1	9.1	3.1888
Gaussian noise addition	5%	2.4	5.9949
	15%	14.5	27
	25%	25.6	38.1696
Sharpening	radius=1	0.9	0.9885
	radius=5	2.4	1.7217
	radius=10	3.2	2.0089
JPEG	90	1.6	0.9566
	70	4.2	4.24
	50	11.8	8.0676
Cropout	0.1	7.7	2.1365
	0.2	13.1	5.3253
	0.3	18.8	8.6735

표 6. 최근 연구와의 비교
Table 6. Comparison with recent researches

Attack	Strength	[10]	[11]	[14]	Proposed (s=2.75)
PSNR		-	-	33.5	33.5
JPEG	50	37	25.4	23.8	0.6696
Cropout	0.3	6	7.5	2.7	5.8355
Dropout	0.3	7	8	2.6	4.7194
Crop	0.035	12	0	11	44.1327
Gaussian filtering	$\sigma=2$	4	50	1.4	4.3048

IV. 결론

본 논문에서 워터마크 및 호스트 영상의 해상도와 관련된 계층을 사용하지 않음으로써 워터마크 및 호스트 영상의 해상도에 적응적인 워터마킹 수행하는 딥 러닝 프레임워크를 제안하였다. 이 방법은 호스트 영상의 비가시성을 확보하기 위해 전처리 네트워크에서 호스트 영상은 해상도

를 유지하고 워터마크 영상의 해상도를 증가시키는 방법이다. 또한 워터마크 정보의 강도를 조절할 수 있도록 하여 비가시성과 강인성의 상보적인 관계를 조절할 수 있도록 하였다.

이 네트워크를 학습한 결과 워터마크 강도를 1로 하였을 때 호스트 영상의 PSNR이 40.58dB로 측정되어 비가시성을 보존하였고, 다양한 화소값 변경공격과 기하학적 공격에 대해 높은 워터마크 추출률을 보이며 높은 공격 강인성을 보였다. 또한, 기존의 연구결과와의 비교에서도 워터마킹 분야에서 의미있는 대부분의 공격에서 더욱 우수한 결과를 보였다.

따라서 제안하는 방법은 다양한 공격에 대해 워터마크 비가시성과 공격 강인성을 적절히 조절하여 유용하게 사용될 수 있을 것으로 사료된다. 특히 다양한 해상도의 호스트 영상과 워터마크 데이터에 대해 추가적인 학습없이 사용할 수 있어 그 효용성이 더욱 높을 것으로 생각된다.

참고 문헌 (References)

- [1] I. J. Cox, et al., "Digital watermarking and steganography," Morgan Kaufmann Publisher, 2008.
- [2] X. Kang, J. Huang, Y.Q. Shi, Y. Lin, "A DWT-DFT composite watermarking scheme robust to both affine transform and JPEG compression," IEEE Trans. Circ. Syst. Video Technol. Vol.13, No.8, pp. 776 - 786, 2003.
- [3] J. George, S. Varma and M. Chatterjee, "Color image watermarking using DWT-SVD and Arnold transform," India Conference (INDICON), pp. 1-6, Dec 2014.
- [4] Y. S. Lee, Y. H. Seo, and D. W. Kim, "Blind image watermarking based on adaptive data spreading in n-level DWT subbands," Security and Communication Networks, Vol.2019, Feb., 2019, <http://dx.doi.org/10.1155/2019/8357251>
- [5] C. Li, Z. Zhang, Y. Wang, B. Ma, D. Huang, "Dither modulation of significant amplitude difference for wavelet based robust watermarking," Neurocomputing, Vol.166, pp. 404 - 415, 2015.
- [6] J. Ouyang, G. Coatrieux, B. Chen and H. Shu, "Color image watermarking based on quaternion Fourier transform and improved uniform log-polar mapping," Computers & Electrical Engineering, <http://dx.doi.org/10.1016/j.compeleceng.2015.03.004>. 419-432 (2015).
- [7] R. Mehta, V. P. Vishwakarma, and N. Rajpal, "Lagrangian support vector regression based image watermarking in wavelet domain," in International Conference on SPIN, Noida, Delhi-NCR, India, pp. 854-859, Feb., 2015.
- [8] H. Hu, Y. Chang, and S. Chen, "A progressive QIM to cope with SVD-based blind image watermarking in DWT domain," IEEE China Summit & International Conference on Signal and Information

- Processing, Xi'an, China, pp. 421-425, July 2014.
- [9] H. Kandi, D. Mishra, S.R.S. Gorthi, "Exploring the learning capabilities of convolutional neural networks for robust image watermarking," *Comput. Secur.* Vol. 65, pp. 247-268, 2017.
- [10] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "HiDDeN: hiding data with deep networks," arXiv:1807.09937, July, 2018
- [11] M. Ahmadi, A. Norouzi, S. M. Reza Soroushmehr, N. Karimi, K. Najarian, S. Samavi, and A. Emami, "ReDMark: framework for residual diffusion watermarking on deep networks," arXiv:1810.07248, Dec. 2018.
- [12] S. M. Mun, S. H. Nam, H. Jang, D. Kim, and H. K. Lee, "Finding robust domain from attacks: a learning framework for blind watermarking," *Neurocomputing*, Vol.337, No.14, pp.191-202, April, 2019, <https://doi.org/10.1016/j.neucom.2019.01.067>
- [13] X. Zhong, and F. Y. Shih, "A robust image watermarking system based on deep neural networks," arXiv:1908.11331, Aug., 2019.
- [14] Y. Liu, M. Guo, J. Zhang, Y. Zhu, and X. Xie, "A novel two-stage separable deep learning framework for practical blind watermarking," in *Proceedings of the 27th ACM International Conference on Multimedia*, France, Oct., 2019.
- [15] Bingyang Wen, and Sergul Aydore, "ROMark, a robust watermarking system using adversarial training," arXiv:1910.01221, Oct., 2019.
- [16] Bas, P., Filler, T., Pevny, T., "Break our steganographic system: the ins and outs of organizing BOSS," *International Workshop on Information Hiding*, Springer, pp. 59-70, 2011.
- [17] Dataset of standard 512×512 grayscale test images. URL <http://decasai.ugr.es/cvg/CG/base.htm>

저 자 소 개



이 재 은

- 2019년 2월 : 광운대학교 전자재료공학과 졸업(공학사)
- 2019년 3월 ~ 현재 : 광운대학교 전자재료공학과(공학석사)
- ORCID : <https://orcid.org/0000-0001-9760-4801>
- 주관심분야 : 영상 처리, 딥 러닝, 뉴로모픽 시스템, SoC 설계



서 영 호

- 1999년 2월 : 광운대학교 전자재료공학과 졸업(공학사)
- 2001년 2월 : 광운대학교 일반대학원 졸업(공학석사)
- 2004년 8월 : 광운대학교 일반대학원 졸업(공학박사)
- 2005년 9월 ~ 2008년 2월 : 한성대학교 조교수
- 2008년 3월 ~ 현재 : 광운대학교 전자재료공학과 정교수
- ORCID : <http://orcid.org/0000-0003-1046-395X>
- 주관심분야 : 실감미디어, 2D/3D 영상 신호처리, 디지털 홀로그래프, SoC 설계



김 동 옥

- 1983년 2월 : 한양대학교 전자공학과 졸업(공학사)
- 1985년 2월 : 한양대학교 공학석사
- 1991년 9월 : Georgia 공과대학 전기공학과(공학박사)
- 1992년 3월 ~ 현재 : 광운대학교 전자재료공학과 정교수
- ORCID : <http://orcid.org/0000-0002-4668-743X>
- 주관심분야 : 3D 영상처리, 디지털 홀로그래프, 디지털 VLSI Testability, VLSI CAD, DSP설계, Wireless Communication