디지털 워터마킹을 위한 딥러닝 기반 하드웨어 가속기의 설계

*이재은 **서영호 ***김동욱

광운대학교

*jelee@kw.ac.kr

Design of deep learning based hardware accelerator for digital watermarking

*Lee, Jae-Eun **Seo, Young-Ho ***Kim, Dong-Wook

Kwangwoon University

요약

본 논문에서는 영상 콘텐츠의 지적재산권 보호를 위하여 딥 러닝을 기반으로 하는 워터마킹 시스템 및 하드웨어 가속기 구조를 제안한다. 제안하는 워터마킹 시스템은 호스트 영상과 워터마크가 같은 해상도를 갖도록 변화시키는 전처리 네트워크, 전처리 네트워크를 거친 호스트 영상과 워터마크를 정합하여 워터마크를 삽입하는 네트워크, 그리고 워터마크를 추출하는 네트워크로 구성된다. 이 중 호스트 영상의 전처리 네트워크와 삽입 네트워크를 하드웨어로 설계한다.

1. 서론

영상 콘텐츠를 가공 및 사용하는 응용분야의 증가로 인해 영상 콘텐츠들의 지적재산권 문제가 발생하고 있다. 고부가가치의 콘텐츠인 영상 콘텐츠의 활성화를 위해서는 지적재산권 보호가 매우 중요한데, 그 방법으로 디지털 워터마킹이 가장 활발하게 연구가 되고 있다[1]. 기존에는 결정론적 알고리즘 기반의 워터마킹 방법이 많이 연구되어 왔지만, 최근에는 딥 러닝 기반의 워터마킹 방법에 대한 연구가 많이 진행되고 있다. 또한, 그 성능이 알고리즘 기반의 워터마킹 방법보다 우수한 성능을 보이고 있다[2-3]. 하지만, 딥 러닝은 엄청난 크기의 메모리와 연산량을 요구하기 때문에 실시간 구현이 어렵다. 특히, 영상을 촬영과 동시에 유통하는 생방송 영상의 저작권을 보호할 수 있는, 고속 구현이 가능한 워터마킹 방법이 요구된다.

따라서, 본 논문에서는 딥 러닝을 기반으로 워터마킹 시스템과 하드웨어 가속기 구조를 제안한다. 제안하는 워터마킹 시스템은 Convolutional Neural Network(CNN)으로 구성되어 있어 일반적인 딥 러닝 기반의 하드웨어 가속기 설계로 간주된다.

2. 워터마킹 시스템

Figure 1에 제안하는 워터마킹 시스템 구조를 나타낸다. 이 방법은 호스트 영상과 워터마크 정보를 각각 전처리하는 네트워크, 삽입하는 네트워크, 공격 시뮬레이션, 그리고 추출 네트워크로 구성된다. 호스트 영상의 전처리 네트워크는 호스트 영상을 보존하며 워터마크를 삽입하기 좋은 특징을 출력하고, 워터마크의 전처리 네트워크는 워터마크를 삽입하기 좋게 변형하며 호스트 영상의 해상도와 동일해지도록 업샘플링(upsampling)한다. 두 개의 전처리 네트워크의 출력을 정합(concatenate)하여 삽입 네트워크를 수행하여 워터마킹된 영상을 출력한다. 그리고, 워터마크 추출이 필요할 시 추출 네트워크를 수행하여 삽입했던 워터마크를 추출한다.

본 논문에서는 호스트 영상의 변화에 따른 구현을 목표하고 있다. 추출기는 영상이 유통된 이후의 과정이므로 고려하지 않는다. 또한, 사전에 선정한 워터마크 정보에 대하여 워터마크 전처리 네트워크를 수행하였다는 가정 하에 워터마크 삽입기를 구현하려고 한다. 구현하고자 하는 네트워크 구조는 6개의 컨볼루션 계층(Convolutional layer)을 가졌으며 Table 1에 보이고 있다. 한 계층에는 컨볼루션, 배치정규화(Batch Normalization, BN), 그리고 활성화(Activation) 함수를 포함한다. 모든 층에서 3×3 컨볼루션을 사용하며 간격(stride)은 1이다. 각 네트워크의 마지막 층은 BN을 사용하지 않았으며, 삽입 네트워크의 마지막 층에는 tanh를, 나머지 층에는 ReLU(Rectified Linear Unit)를 사용한다.

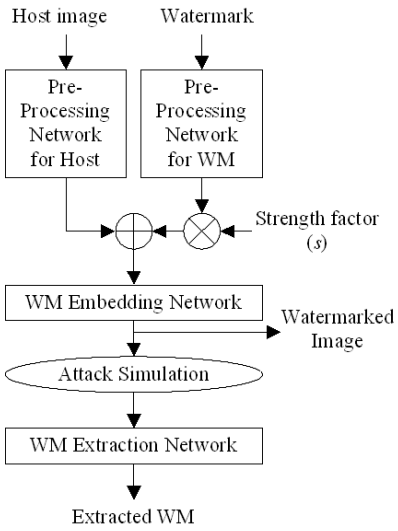

Figure 1. Overall structure of the proposed watermarking system

3. 제안하는 하드웨어 구조

전체적인 하드웨어 구조의 블록도는 Figure 2에 나타내었다. 호스트 영상의 픽셀 값과 미리 학습해둔 네트워크의 가중치가 입력으로 들어가면 CONV 부분을 먼저 수행한다. 그 다음 누적 덧셈이 필요한 값들은 내부 메모리에 저장해둔 다음에 누적 덧셈을 수행한다. 그리고, BN과 활성화 함수를 수행하기 위해 POST 부분이 존재한다. CONV 부분에서는 입력으로 들어온 가중치 36개와 IFM 36개의 곱셈 및 덧셈으로 3×3×4의 컨볼루션을 수행하여 1개의 값을 출력한다. POST 부분에서는 BN과 활성화 함수를 수행하는데, Table 1에 보였듯이 연산하는 층마다 다른 구성을 갖는다. 첫번째 계층은 BN과 활성화 함수를 수행하지 않아 입력을 그대로 출력한다. 중간 계층은 BN과 활성화 함수로 ReLU를 수행한다. ReLU는 최상위 비트를 확인하여 1이면 0으로, 0이면 입력을 그대로 출력하는 MUX를 사용하여 구현하였다. 마지막 계층은 BN을 수행하지 않고 활성화 함수로 tanh를 수행한다. 지금까지의 연산은 곱셈기와 덧셈기로 가능하지만, 마지막 층에서의 tanh 활성화 함수는 지수 연산과 나눗셈의 연산을 필요로 한다. 하지만, 지수 연산과 나눗셈 연산은 비교적으로 굉장히 큰 연산량을 요구하기 때문에 look-up table로 대체하였다. 제안하는 하드웨어 모듈을 구현하기 위해, 레지스터로 258Bytes, 내부 메모리로 43,120Bytes DPRAM이 필요하다. 입력을 위한 DPRAM이 2,160Bytes, 중간 결과 저장을 위한 DPRAM이 40,960Bytes가 요구된다.

4. 결과

워터마킹 시스템의 비가시성과 강인성은 호스트 영상과 원본 영상의 PSNR은 37.6775(dB)이고 추출한 워터마크와 원본 워터마크의 BER은 0.6696(%)로, 매우 강인하고 비가시성을 보존하는 시스템이라고 볼 수 있다. 성능을 검증한 후에, Synopsys의 VCS를 이용하여 verilog로 구현하고 시뮬레이션을 진행하였다. 그 결과를 Figure 3에 보이고 있다. Fig. 3(a)는 일정 시간이 지나고 입력 신호가 들어오며 Mul_s가 1이 되었을 때 곱셈의 결과를 확인할 수 있으며 Fig. 3(b)는 POST 부분의 덧셈 결과를 확인할 수 있다.

Table 1. Network structure to be implemented

Network	Kernel size	number of kernels	stride	Batch Normalization	Activation function
Pre-processing Network	3×3	64	1	X	X
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	X	tanh

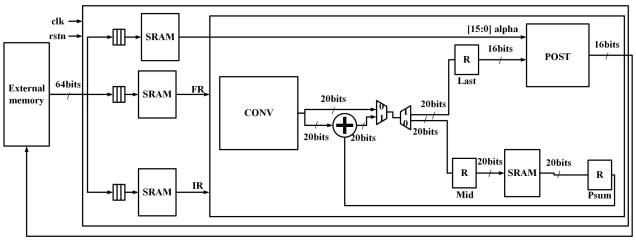

Figure 2. Block diagram of the proposed hardware structure

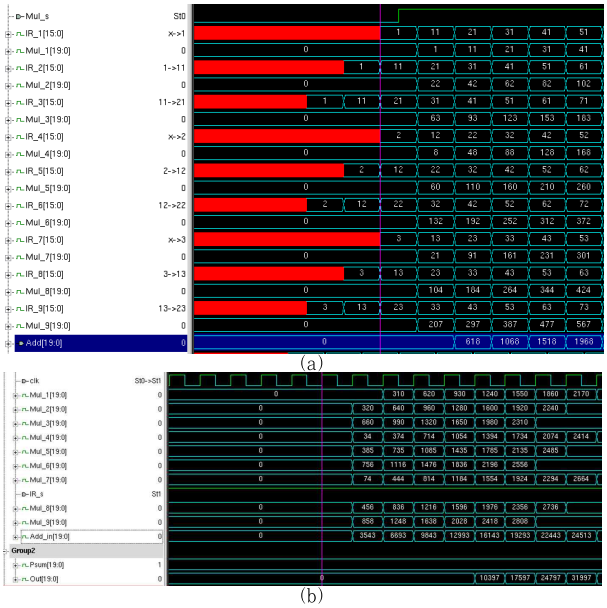

Figure 3. Simulation results for our hardware: (a) input data and multiplication output, (b) outputs

5. 결론

본 논문에서는 딥 러닝을 기반으로 하는 워터마킹 시스템 및 하드웨어 가속기 구조를 제안한다. 워터마크 데이터는 사전에 수행하였다는 가정 하에 새로운 호스트 영상의 전처리 네트워크와 워터마크 삽입기를 구현하였다. 곱셈기와 덧셈기로 컨볼루션 연산과 배치 정규화의 구현을, 최상위비트만을 확인하여 ReLU 활성화 함수를, 그리고 지수 함수 연산과 나눗셈 연산을 요구하는 tanh 활성화 함수는 look-up table을 이용하여 계산 시간을 단축하였다.

본 회로설계는 고속화로 호스트 영상이 유통되는 경우에 바로 적용시킬 수 있을 것이라 생각된다. 또한, CNN을 디지털 회로로 구현함으로서 딥 러닝의 하드웨어 가속기로서 추후 연구에 도움이 될 것이다.

감사의 글

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)

참고문헌

[1] I. J. Cox, et al., "Digital watermarking and steganigraphy," Morgan Kaufmann Publisher, 2008.
[2] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "HiDDeN: hiding data with deep networks," arXiv:1807.09937, July, 2018.
[3] 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.