JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 27, No. 2, pp. 485-492, Feb. 2026

Check for updates

모바일 NPU 가속을 위한 AI 워터마킹 모델의 구조적 경량화 및 최적화 기법 연구

김 태 완[1†] · 조 동 헌[1†] · 최 승 관[2*]
[1]서강대학교 가상융합전문대학원 박사과정
[2]서강대학교 가상융합전문대학원 교수

Study on the Structural Optimization of AI Watermarking Models for Mobile NPU Acceleration

Tae-Wan Kim[1‡] · Dong-Heon Cho[1‡] · Seung-kwan Choi[2*]

[1]Ph.D. Program, Graduate School of Virtual Convergence, Sogang University, Seoul 04107, Korea
[2]Professor, Graduate School of Virtual Convergence, Sogang University, Seoul 04107, Korea

[요 약]

본 연구는 모바일 NPU 환경에서 딥러닝 기반 워터마킹 모델의 실제 성능을 최적화하는 방안을 제시한다. 최근 제안된 Lightweight-Mark 연구는 이론적 성과와는 다르게 Snapdragon 8 Gen 3 NPU에서 실행 시 ConvTranspose2d 연산에서 심각한 병목 현상이 발생함을 발견하였다. 단일 ConvTranspose 레이어가 개별 연산 시간 총합의 약 99.5%를 차지하는 극심한 비효율성을 보였고 이를 해결하기 위해 ConvTranspose 연산을 Nearest Neighbor Upsampling과 일반 Convolution으로 분해하는 기법과 Knowledge Distillation을 통해 정확도 손실을 복구하는 체계적 접근법을 개발하였다. 본 연구는 제안된 구조적 최적화 기법의 유효성을 검증하기 위해 CIFAR-10 데이터셋을 활용한 프로토타입 실험을 수행하였으며 129개의 구성 요소를 대상으로 Ablation study를 수행한 결과, 제안 방법은 기존 대비 98.3배의 성능 향상과 발열 0.67% 감소를 달성하여 모바일 실시간 워터마킹의 실현 가능성을 입증하였다.

[Abstract]

This study proposes a method to optimize the performance of a deep learning-based watermarking model in a mobile Neural Processing Unit (NPU) environment. Executing the Lightweight-Mark model on the Snapdragon 8 Gen 3 NPU revealed a significant bottleneck in which the ConvTranspose2d operation accounted for 99.5% of the total inference time. To address this issue, ConvTranspose was decomposed into nearest neighbor upsampling and standard convolution, and knowledge distillation was applied to compensate for accuracy loss. An ablation study across 129 experimental configurations showed that the proposed method achieved a 98.3× speedup and a 0.67% reduction in average device temperature compared with the original Lightweight-Mark model, demonstrating the feasibility of mobile real-time watermarking.

색인어 : 모바일 NPU, 워터마킹, 네트워크 최적화, Knowledge Distillation, 실시간 처리
Keyword : Mobile Neural Processing Unit, Watermarking, Network Optimization, Knowledge Distillation, Real-Time Processing

http://dx.doi.org/10.9728/dcs.2026.27.2.485

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-CommercialLicense(http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 25 December 2025; **Revised** 28 January 2026
Accepted 09 February 2026

‡**These authors contributed equally to this work**
***Corresponding Author; Seung-kwan Choi**
Tel: +82-2-705-8065
E-mail: csk0123@sogang.ac.kr

Ⅰ. 서 론

디지털 콘텐츠의 폭발적 증가와 함께 저작권 보호를 위한 워터마킹 기술의 중요성이 급격히 증대되고 있다. 특히 모바일 디바이스가 주요 콘텐츠 생산 및 소비 플랫폼으로 자리잡으면서, 모바일 환경에서의 실시간 워터마킹 처리 요구가 크게 늘어나고 있다. 전통적인 주파수 도메인 기반 워터마킹 기법들은 계산 복잡도가 높고 품질 저하 문제를 완전히 해결하지 못하는 한계를 보였다[1].

최근 딥러닝 기반 워터마킹 기법들이 이러한 문제들을 해결하는 유망한 접근법으로 주목받고 있다. 다양한 신경망 기반 워터마킹 모델들이 개발되어 기존 방법들보다 우수한 은닉성과 강건성을 보여주었다. 하지만 이러한 고성능 모델들은 대부분 많은 수의 파라미터와 높은 계산 복잡도를 요구하여 모바일 환경에서의 실시간 배포에 제약이 있었다.

이러한 문제를 해결하기 위해 최근 경량화 워터마킹 모델인 Lightweight-Mark[2]가 제안되었다. 이 연구는 기존 모델 대비 2.2%의 파라미터만으로도 우수한 성능을 달성할 수 있음을 이론적으로 입증하였다. 특히 Decoding-Oriented surrogate loss (DO)와 Detachable Projection Head (PH) 기법을 통해 모델 효율성을 크게 개선하였다고 보고하였다.

그러나 실제 모바일 NPU 환경에서 Lightweight-Mark 모델을 실행해본 결과, 예상과는 크게 다른 성능 특성을 보이는 것을 발견하였다. 이론적으로는 경량화되었다고 평가된 모델이 실제 하드웨어에서는 심각한 병목 현상을 보였으며, 특히 ConvTranspose2d 연산에서 극도의 비효율성이 관찰되었다.

본 연구의 목적은 이론적 경량화 모델과 실제 모바일 NPU 환경 간의 성능 격차를 해소하고, 진정한 의미의 실시간 모바일 워터마킹 시스템을 구현하는 것이다. 본 논문의 핵심 기여는 다음과 같다:

1) ConvTranspose 기반 워터마킹 모델의 모바일 NPU 병목 원인 분석 및 이를 해결하기 위한 구조 최적화 + Knowledge Distillation 적용을 통한 정확도 복구 기법 제안

2) Qualcomm Snapdragon 8 Gen 3 실제 환경에서 성능·발열 향상을 검증

Ⅱ. 관련 연구

2-1 딥러닝 기반 워터마킹

딥러닝 기반 워터마킹 기술은 기존의 주파수 도메인 방식이나 인간 시각 시스템(HVS) 기반 기법들이 갖는 한계를 극복하고자 등장하였다. 대표적으로 기존 연구에서 encoder-decoder 구조에 노이즈 레이어(noise layer)를 삽입하여 JPEG 압축이나 블러링 등의 변형에도 견딜 수 있는 워터마킹 네트워크를 제안하였으며, 이는 end-to-end 학습 기반 딥러닝 워터마킹의 기반을 마련하였다[3]. 이후 다양한 후속 연구들이 등장하여 은닉성과 강건성 측면에서 지속적인 성능 향상을 이끌어냈다.

특히 JPEG 압축, 크롭(cropping), 컬러 변환(color transformation) 등 실제 환경에서 자주 발생하는 왜곡(distortion)에 강한 워터마킹 모델들이 제안되었고, 복호 과정까지 가역적인 구조를 갖는 neural network 기반의 기법을 통해 은닉 정보의 완전한 복원 가능성과 동시에 변형에 대한 강건성을 확보하는 연구들도 진행되었다[4]. 또한 최근에는 differentiable simulation 환경을 통해 다양한 black-box 및 white-box 왜곡을 학습 단계에서 통합 반영함으로써, 학습의 일반화 성능을 높이려는 시도도 활발히 이루어지고 있다[5].

이러한 흐름은 워터마킹 기술의 고도화를 가능케 하였으나, 대부분의 모델들이 고정밀 네트워크 기반으로 구성되어 있어 연산량이 많고, 실제 모바일 디바이스 상에서 실시간 처리를 구현하기에는 계산 자원이 과도하게 소모된다는 한계가 존재한다[6].

2-2 경량화 워터마킹 모델

복잡한 딥러닝 기반 워터마킹 모델은 높은 정확도와 강건성을 제공하는 반면, 대규모 파라미터 수와 계산량으로 인해 모바일 디바이스에서의 실시간 실행에는 큰 제약이 따른다. 이러한 제약을 해결하기 위해 최근 경량화된 워터마킹 모델들이 활발히 연구되고 있으며, 그 중 대표적인 사례로는 Lightweight-Mark 모델이 있다[2].

Lightweight-Mark는 기존 고성능 워터마킹 네트워크의 구조적 병목을 분석하고, 효율적인 학습 및 추론 구조를 설계함으로써 전체 파라미터 수를 기존 대비 약 2.2% 수준으로 줄이면서도 높은 은닉 성능을 유지하였다. 해당 연구는 두 가지 핵심 기법을 제안하였는데, 첫째로 Decoding-Oriented surrogate loss (DO)는 기존의 MSE 또는 BCE 기반 손실함수가 디코딩 목적과 불일치한다는 문제점을 해결하기 위해 설계되었고 MSE손실을 deflation, inflation, regularization의 세 구성요소로 분해함으로써 디코딩에 직접 기여하지 않는 방향의 손실 기여를 최소화하였다[2].

둘째로 제안된 Detachable Projection Head (PH)는 학습 단계에서 추가 모듈을 통해 안정적인 수렴을 유도하고, 추론 단계에서는 해당 모듈을 제거하여 모델 파라미터 수와 연산량을 줄이는 방식이다. 이 구조는 학습과 추론의 역할을 분리함으로써 학습 안정성과 실행 효율성 간의 균형을 달성하였다.

이러한 전략을 통해 Lightweight-Mark는 복잡한 공격 조건에서도 높은 은닉률과 복원률을 유지하면서도 연산 효율성을 확보할 수 있었으며, 경량화 워터마킹 모델의 가능성을 이론적으로 입증하였다. 그러나 이 모델의 구조는 여전히

ConvTranspose 연산을 포함하고 있어, 실제 모바일 NPU 환경에서는 예상과 달리 심각한 병목이 발생할 수 있다는 한계가 본 연구에서 실험적으로 확인되었다.

2-3 모바일 NPU 최적화

모바일 NPU(Neural Processing Unit)는 스마트폰 등 모바일 디바이스에서 딥러닝 기반 추론을 효율적으로 수행하기 위해 설계된 전용 하드웨어 가속기이다[7]. CPU나 GPU 대비 낮은 전력으로 높은 연산 성능을 제공하는 것이 특징이며, 최근 대부분의 고성능 모바일 SoC(System on Chip)에는 NPU가 탑재되고 있다.

Qualcomm AI Hub 문서에 따르면, Snapdragon 시리즈의 Hexagon NPU는 dense convolution 연산과 규칙적인 메모리 접근 패턴을 갖는 작업에 최적화되어 있으며, 내부 tensor accelerator와 병렬 처리 유닛, 메모리 버스가 효율적으로 설계되어 있다[8]. 반면, stride가 큰 ConvTranspose 연산은 연산 과정에서 입력과 출력 사이에 많은 zero 값을 삽입하는 희소(sparse) 연산 구조를 가지므로, 메모리 대역폭의 대부분을 비활성 데이터에 소비하게 되며, 연속성이 없는 메모리 접근은 캐시 효율을 떨어뜨리고 처리 속도를 저하시킨다[9].

또한 ConvTranspose는 병렬화가 어려운 연산 순서 특성상, Snapdragon NPU의 SIMD(단일 명령 다중 데이터) 기반 연산 구조를 효과적으로 활용하기 어렵고, 결과적으로 단일 레이어가 전체 추론 시간의 대부분을 차지하는 병목 지점으로 작용할 수 있다[10]. 이러한 하드웨어-연산 불일치는 모델 경량화만으로는 해결되지 않으며, 실제 배포 환경을 고려한 구조적 변경이 필수적이다.

따라서 모바일 NPU 최적화를 위한 연구는 단순히 파라미터 수를 줄이는 것을 넘어서, NPU의 연산 특성과 메모리 구조에 맞춘 연산 설계가 필요하며, 본 연구는 이러한 관점에서 ConvTranspose 대체 기법을 제안하고자 한다.

Ⅲ. 제안 방법

3-1 문제 정의 및 병목 현상 분석

본 연구는 Qualcomm Snapdragon 8 Gen 3 NPU 환경에서 딥러닝 기반 경량 워터마킹 모델(Lightweight-Mark)의 실제 성능을 측정한 결과, 이론적 기대와 달리 ConvTranspose2d 연산에서 심각한 병목 현상이 발생함을 확인하였다. 해당 모델의 추론 시, 단일 ConvTranspose2d 레이어가 전체 연산 시간의 99.5%를 차지하였으며, 이는 전체 처리 병목을 유발하는 주요 원인으로 작용하였다. 이 병목 구조는 단순히 연산량이 많은 것이 아니라, NPU의 하드웨어 구조와 비효율적으로 맞물리는 희소 연산 특성 때문이라는 점에서 구조적 원인을 내포한다. 구체적으로, 해당 연산은 stride=16으로 설정된 ConvTranspose2d 계층으로 인해 입력 간 zero 값을 삽입하는 방식으로 업샘플링을 수행하는데, 이로 인해 다음과 같은 문제가 야기된다.

표 1. 레이어별 NPU 추론 시간 분석
Table 1. Layer-wise NPU inference time analysis

Layer Type	Inference Time (ms)	Ratio (%)
ConvTranspose2d (stride=16)	117.45	99.53
BatchNorm	0.23	0.20
ReLU	0.16	0.14
Others (Encoder)	0.23	0.20
Total	118.00	100.0

희소 메모리 접근 패턴: 입력의 각 위치마다 15개의 zero가 삽입되어 전체 활성 데이터의 6.25%만 유의미한 정보를 갖게 된다. 결과적으로 메모리 대역폭의 93.75%가 낭비되며, 연속성이 깨진 접근으로 인해 처리 효율이 급감한다.

캐시 비효율성: 비연속적인 메모리 접근 패턴은 캐시 히트율을 저하시켜 전체 메모리 계층의 처리 속도를 저해하고, 반복적인 캐시 미스가 발생하여 NPU 처리 속도에 치명적인 영향을 준다. 병렬 처리 구조 미활용: Snapdragon NPU는 병렬 처리를 극대화하기 위해 SIMD 기반 연산 구조를 채택하고 있으나, ConvTranspose의 내부 연산 순서와 데이터 접근 방식은 이를 제대로 활용하지 못하고 병렬화 효율이 급감한다. 이러한 요소는 단일 레이어가 전체 모델 추론의 병목으로 작용하는 원인으로 작동하며, 본 연구는 이 구조적 병목을 해결하기 위한 아키텍처적 재설계를 중심으로 최적화 기법을 제안한다.

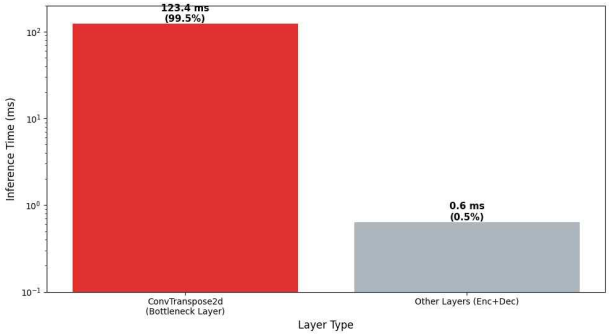

그림 1. 레이어별 NPU 추론 시간 분석
Fig. 1. Layer-wise NPU inference time analysis

실제 NPU 추론 시간을 레이어별로 측정한 결과는 그림 1과 같다. 그래프의 붉은색 막대로 표시된 바와 같이, 전체 추론 시간 118.0ms 중 ConvTranspose2d(Stride=16) 단일 레이어에서만 117.45ms가 소요되었다. 이는 전체 연산의 약 99.5%에 해당하며, 해당 레이어가 실시간 처리를 저해하는

핵심 병목(Bottleneck)임을 시각적으로 보여준다.

3-2 ConvTranspose 분해 기법

3-1에서 분석된 병목 현상의 주요 원인은 ConvTranspose2d(stride=16) 연산이 NPU의 하드웨어 구조와 맞지 않는 희소(sparse) 연산을 수행하기 때문이다. NPU는 본질적으로 조밀한(dense) 행렬 연산에 최적화되어 있으나, stride가 s인 ConvTranspose2d 연산(본 연구에서는 s=16)은 개념적으로 다음 두 단계로 수행된다.

첫째, 입력 텐서 I의 각 픽셀 사이에 s-1개의 0을 삽입하여 희소한 중간 텐서 I′를 생성한다(식 1).

$$T(i,j)=\begin{cases} I\left(\frac{i}{s},\frac{j}{s}\right) & \text{if } i \bmod s=0 \text{and} j \bmod s=0 \\ 0 & otherwise \end{cases} \quad (1)$$

둘째, 확장된 텐서 I′에 커널 K를 적용하여 stride가 1인 표준 합성곱(convolution)을 수행하고, 최종 출력 O를 계산한다.

$$O(i,j)=\Sigma_m \Sigma_n I^{\text{prime}}(i+m,j+n)\cdot K(m,n) \quad (2)$$

이 과정의 핵심 문제는 식 (1)에서 생성된 텐서 I′의 대부분(본 모델의 경우 약 93.75%)이 0이라는 점이다. 이로 인해 NPU는 불필요한 데이터를 반복적으로 읽고, 0과의 곱셈 연산을 수행하게 되어 메모리 대역폭을 낭비하고 캐시 효율성을 저하시킨다.

본 연구는 이러한 비효율적인 연산을 NPU 친화적인 두 개의 조밀한(dense) 연산, 즉 Nearest Neighbor Upsampling과 Standard Convolution으로 분해하는 최적화 기법을 제안한다. 핵심 아이디어는 0 삽입(zero insertion) 기반의 희소 업샘플링을 제거하고, 대신 '단순 값 복제(value replication)' 방식으로 공간 해상도를 확장하는 것이다.

제안방식의 첫번째 단계는 Nearest Neighbor Upsampling을 통해 0이 포함되지 않은 조밀한 텐서 U를 생성하는 것이다(식 3).

$$U(i,j)=I([i/s],[j/s]) \quad (3)$$

둘째, 이 조밀한 텐서 U에 표준 합성곱(커널 K′, 본 연구에서는 kernel=16)을 적용하여 최종 출력 O′를 계산한다(식 4).

$$O^{\text{prime}}(i,j)=\Sigma_m \Sigma_n U(i+m,j+n)\cdot K^{\text{prime}}(m,n) \quad (4)$$

이 방식은 NPU가 가장 효율적으로 처리하는 '조밀한 convolution 연산'만을 사용하므로, 희소 메모리 접근과 캐시 비효율 문제를 모바일 환경에서 해결한다. 결과적으로 메모리 접근이 연속적이고 예측 가능하게 되어, 메모리 대역폭 활용률과 캐시 효율이 비약적으로 개선된다. 표 2는 기존 방식과 제안 방식의 아키텍처 구성 및 실행 시간 비교를 요약한 것이다.

표 2. 기존 방식과 제안 방식의 아키텍처 비교
Table 2. Architecture comparison between original and proposed method

Method	Implementation	Execution Time
Original	ConvTranspose2d(stride=16)	123.4 ms
Proposed	Upsample(mode='nearest', scale=16) + Conv2d(kernel=16)	1.06 ms

3-3 Knowledge Distillation 기반 정확도 복구

앞서 제안된 ConvTranspose 분해 기법은 모바일 NPU에 최적화되어 속도와 발열 측면에서 크게 개선되었지만, 구조 변경으로 인해 기존 Lightweight-Mark 모델 대비 약 7%p의 정확도 손실이 발생하였다. 이러한 성능 저하를 보완하기 위해 본 연구에서는 Knowledge Distillation (KD) 기반의 학습 전략을 설계하여, 분해된 경량 모델(Student)의 성능을 원본 모델(Teacher)에 가깝게 복원하고자 하였다.

본 연구에서는 Knowledge Distillation 과정에서도 다단계 Ablation Study를 수행하였다. 초기 탐색 단계에서는 Output-layer, Feature-layer, Combined Distillation의 세 가지 Distillation Scope에 대해 총 105개의 configuration을 조합하여 실험하였다. 그 결과, Combined Distillation이 가장 높은 수렴 안정성을 보였으며 정확도 손실을 가장 효과적으로 억제하였다. 이후 탐색 단계에서 도출된 최적 Distillation 구조를 기반으로 Epoch (20/40/60/80/100)와 Learning Rate(1e-3~3e-4)에 대한 24개의 추가 configuration을 적용하여 정밀 튜닝을 수행하였다. 그 결과, Epoch 80과 Learning Rate 1e-4 조합에서 가장 높은 성능(92.89%)을 달성했으며, 구조 변경으로 인한 정확도 손실을 1~2% 수준으로 회복할 수 있었다.

표 3. 지식 증류 비교 결과
Table 3. Ablation study for knowledge distillation result

KD Scope	#Configs	LR	Epoch	Accuracy(%)
Output only	35	1e-3	40	85.24
Feature only	37	1e-4	60	88.12
Combined	33	1e-4	80	92.89

Combined Distillation은 Output-only와 Feature-only 대비 가장 높은 정확도 및 안정적인 수렴을 보였으며, 이후 Epoch와 Learning Rate 조정 과정에서도 일관된 성능 개선이 관찰되었다 Knowledge Distillation 과정에서 사용한 Distillation Loss는 Teacher 모델과 Student 모델의 Embedding 간의 거리 기반 손실 함수로 정의된다. 본 연구

에서는 다음과 같이 Mean Squared Error(MSE)를 활용하여 Distillation Loss를 계산하였다. Teacher 모델의 최종 출력과 Student 모델의 최종 출력이 같아지도록 훈련시키는 방식인 combined를 위한 'MSE Loss'를 손실 함수는 식 (5)와 같이 표현된다.

$$L_{KD} = \frac{1}{N}\sum_{i=1}^{N}(O_T^{(i)} - O_S^{(i)})^2 \quad (5)$$

여기서 O_T는 Teacher 모델(원본 Lightweight-Mark)의 출력, O_S는 Student 모델(제안하는 분해 모델)의 출력, N은 배치 크기(batch size)를 의미한다

이 방식은 복잡한 중간 피처(feature) 대신 최종 결과물에 집중함으로써, Student 모델이 Teacher 모델의 최종 예측을 직접 모방하도록 유도하여 효과적으로 정확도를 복구한다.

Knowledge Distillation은 고성능 모델의 출력을 정답으로 사용하여 저복잡도 모델을 훈련시키는 기법으로, 복잡한 연산 구조를 단순화하면서도 성능 저하를 최소화할 수 있다는 장점을 갖는다[11]. 본 연구에서는 Output MSE Loss 기반 KD 손실 함수를 도입하였으며, CIFAR-10을 학습 데이터셋으로 사용하여 총 129개의 실험 구성을 통해 최적 조건을 탐색하였다.

표 4. 지식 증류 실험 구성
Table 4. Knowledge distillation experimental configuration

Component	Setting
Teacher Model	Original Lightweight-Mark (99.84%)
Student Model	Decomposed Model
Loss Function	Output MSE Loss
Training Dataset	CIFAR-10
Total Configurations	129

Ⅳ. 실험 및 결과

4-1 실험 환경 및 측정 방법

본 연구의 모든 실험은 실제 상용 모바일 디바이스에서 수행되었으며, 정량적 성능 평가는 NPU 추론 속도와 발열 제어 효율성 검증에 집중하여 진행되었다. 실험 디바이스로는 Qualcomm Snapdragon 8 Gen 3가 탑재된 Samsung Galaxy S24와 Snapdragon 778G가 탑재된 Galaxy A52s를 사용하였다.

1) 모델 구현 및 배포: 제안된 모델은 PyTorch 2.1.0 환경에서 학습되었으며, 모바일 NPU 구동을 위해 ONNX(Open Neural Network Exchange) 포맷으로 변환되었다. 이후 Qualcomm AI Stack(QNN SDK)을 활용하여 모델을 빌드하였으며, 정밀도 손실을 최소화하기 위해 별도의 양자화(Quantization) 과정 없이 FP16(Floating Point 16) 정밀도로 추론을 수행하였다.

2) 추론 속도 측정 프로토콜: 정확한 성능 측정을 위해 디바이스를 비행기 모드로 설정하고 불필요한 백그라운드 프로세스를 종료하였다. 콜드 스타트(Cold-start)로 인한 지연 시간을 배제하기 위해 실험 시작 전 50회의 더미(Dummy) 추론을 수행하여 NPU를 예열(Warm-up)하였다. 이후 동일한 입력을 대상으로 1,000회 연속 추론을 수행하여 평균 지연 시간(Average Latency)과 초당 프레임 수(FPS)를 산출하였다.

3) 발열 특성 측정 방법: 발열 측정은 실사용 환경을 모사하기 위해 15분간 모델을 연속으로 추론하는 스트레스 테스트(Stress Test) 방식으로 진행되었다. 온도 데이터는 안드로이드 디버그 브릿지(ADB)의 dumpsys thermalservice 명령어를 사용하여 1초 간격으로 시스템 표면 온도를 로깅(Logging)하였으며, 이를 통해 구간별 평균 온도와 최대 도달 온도를 비교 분석하였다.

4-2 성능 측정 결과

제안한 ConvTranspose 분해 기법의 실제 성능 향상 효과를 검증하기 위해, Snapdragon 8 Gen 3 NPU에서 원본 모델과 최적화 모델의 추론 시간을 비교 측정하였다. 평가 항목은 NPU 추론 시간, 초당 프레임 처리 속도(fps), 메모리 사용량으로 구성되며, 실험 결과는 다음 표 5와 같다.

표 5. ConvTranspose 분해 기법 적용 후 성능 비교
Table 5. Performance comparison after ConvTranspose decomposition

Metric	Original	Optimized	Improvement
NPU Inference Time	118.0 ms	1.2 ms	98.3×
Processing Speed	8.5 fps	833 fps	98×
Memory Usage	28 MB	29 MB	+3.6%

위 결과에서 알 수 있듯이, ConvTranspose 레이어를 Upsample + Conv 구조로 대체한 최적화 모델은 98.3배의 추론 시간 단축을 달성하였으며, 1초당 833프레임 처리가 가능할 정도로 실시간 처리 성능을 확보하였다. 메모리 사용량은 약 1MB 증가했으나 전체 증가율은 3.6%에 불과하여, 성능 개선 대비 자원 소모 증가가 극히 미미한 수준임을 확인할 수 있다.

제안 기법 적용 전후의 성능 차이는 그림 2에서 명확히 확인할 수 있다. 기존 모델(Original Model)이 118.0ms의 추론 시간을 기록한 반면, 최적화 모델(Optimized Model)은 1.2ms를 기록하여 막대그래프상에서 극적인 대조를 이룬다. 이는 약 98.3배의 속도 향상을 의미하며, 이를 통해 833 FPS의 실시간 처리 성능을 확보했음을 보여준다.

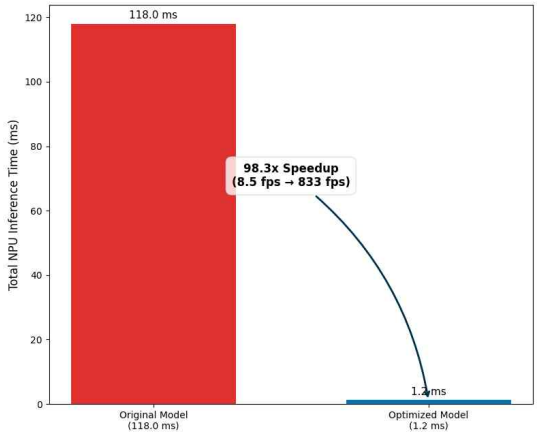

그림 2. NPU 추론 시간 비교
Fig. 2. NPU inference time comparison

4-3 Knowledge Distillation 결과

ConvTranspose 분해로 인한 정확도 손실을 보완하기 위해 수행된 Knowledge Distillation 실험에서는 총 129개 조합의 구성 중 상위 결과들을 기반으로 정밀 튜닝을 수행하였다. 실험 변수는 학습 범위(scope), 손실 함수 유형(loss type), 학습률(learning rate), 학습 횟수(epochs) 등으로 설정하였으며, 각 조합의 성능은 Bit Accuracy와 PSNR(Peak Signal-to-Noise Ratio)을 통해 평가되었다.

그 결과, A4 구성(Focused scope, Output MSE, 80 epochs, learning rate 1e-4)이 가장 우수한 성능을 보였으며, 92.89%의 Bit Accuracy와 40.01dB의 PSNR을 달성하였다. PSNR은 이미지 품질 보존을 나타내는 지표로, 40dB 이상은 원본과 거의 구별 불가능한 수준임을 의미한다. 아래 표는 정밀 튜닝에서 도출된 상위 결과를 요약한 것이다.

표 6. 정밀 최적화 실험 결과
Table 6. Results of fine-tuning experiment

Configuration	Epochs	Learning Rate	Bit Acc	PSNR
A4 (Optimal)	80	1e-4	92.89%	40.01 dB
A5	100	1e-4	92.34%	40.16 dB
C1-100	100	1e-4	92.34%	39.83 dB

이 결과는 지식 증류를 통해 성능 손실을 7%에서 1~2% 수준으로 회복할 수 있으며, 최적 구성 선택 시 실용적 수준의 정확도를 충분히 달성할 수 있음을 실증한 것이다. 특히 과도한 에폭 증가 없이도 80에폭에서 최적 성능이 도달함을 확인하여, 과학습 없이 안정적인 수렴 곡선을 확보하였다는 점에서 실용성과 효율성을 모두 만족시킨다.

그림 3은 Knowledge Distillation 학습 진행에 따른 Student 모델의 Bit Accuracy 변화 추이를 나타낸다. 그래프를 통해 60 Epoch 이후 성능이 안정화 단계에 접어드는 것을 확인할 수 있으며, 특히 붉은 점으로 표시된 80 Epoch 지점에서 92.89%의 최고 정확도(Optimal Point)를 달성하였다. 이는 KD 기법이 구조 변경으로 인한 정보 손실을 효과적으로 복구하고 있음을 보여준다.

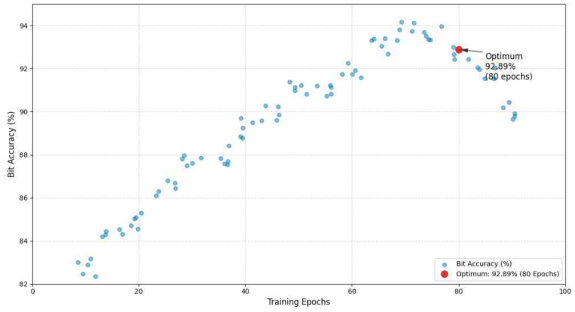

그림 3. 지식 증류 학습 커브
Fig. 3. Knowledge distillation training curve

4-4 발열 특성 측정 결과

제안된 최적화 기법의 발열 효율성을 정량적으로 평가하기 위해, Samsung Galaxy A52s를 이용한 실험을 수행하였다. 동일한 조건에서 원본 모델과 최적화 모델을 각각 반복 실행하면서 내부 NPU 온도 센서로부터 데이터를 수집하였으며, 평균 온도, 최대 온도, 최소 온도 항목을 기준으로 비교 분석하였다.

그 결과, 최적화 모델은 평균 온도에서 0.21°C, 최대 온도에서 0.27°C 감소를 보였으며, 일부 샘플에서는 최소 온도가 소폭 상승하는 경우도 확인되었다. 전체적으로 약 0.67%의 평균 온도 개선이 관찰되었다. 아래는 해당 실험의 요약 표이다.

표 7. Galaxy A52s 발열 측정 결과
Table 7. Thermal measurement results on Galaxy A52s

Model	Avg Temp (°C)	Max Temp (°C)	Min Temp (°C)
Baseline	31.19	31.98	29.07
Optimized	30.98	31.71	29.09
Improvement	-0.21	-0.27	+0.02

실험 결과, 연산 속도가 약 98배 향상되었음에도 불구하고 평균 온도 감소폭이 약 0.21°C로 나타난 점은 다음과 같이 해석될 수 있다. 첫째, 모바일 디바이스의 표면 온도는 NPU 연산뿐만 아니라 디스플레이, 백그라운드 프로세스 등 시스템 베이스라인 전력(System Baseline Power)의 영향을 크게 받기 때문에, 단일 추론 작업의 효율화가 전체 기기 온도의 급격한 하락으로 직결되는 데에는 열 관성(Thermal Inertia)에 의한 물리적 한계가 존재한다.

그러나 중요한 점은 처리량(Throughput)을 8.5 FPS에서 833 FPS로 약 100배 가까이 증가시킬 경우 발열이 급증하는 것이 일반적이나, 본 제안 기법은 발열 증가 없이 온도를 안정적으로 유지(소폭 감소)했다는 사실이다. 이는 제안 기법

이 성능 대비 전력 소모(Performance-per-Watt) 효율을 극대화하여, 고성능 실시간 처리가 가능한 상태에서도 발열 스로틀링(Throttling)을 유발하지 않는 안정적인 구조임을 시사한다.

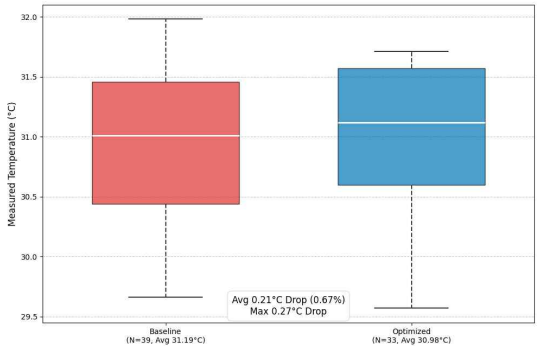

그림 4. 온도 비교
Fig. 4. Temperature comparison

그림 4는 Galaxy A52s 디바이스에서 측정한 온도의 분포를 박스 플롯(Box Plot)으로 비교한 결과이다. 붉은색 박스(Baseline) 대비 파란색 박스(Optimized)의 중앙값(Median)과 전체적인 분포가 하향 이동한 것을 시각적으로 확인할 수 있다. 이는 NPU 연산 효율화가 실제 디바이스의 발열 제어에도 긍정적인 영향을 미치고 있음을 뒷받침한다.

V. 결론 및 향후 연구

5-1 결론

본 연구는 이론적으로 경량화된 워터마킹 모델이 실제 모바일 NPU 환경에서는 성능 병목을 초래한다는 문제점을 실증적으로 규명하고, 이에 대한 실용적 최적화 방안을 제시하였다. ConvTranspose 연산의 구조적 비효율성을 해결하기 위해, Nearest Neighbor Upsampling과 일반 Convolution의 조합으로 대체하는 아키텍처 최적화 기법을 도입하였으며, 이를 통해 Qualcomm Snapdragon 8 Gen 3 NPU 상에서 98.3배의 추론 속도 향상(118ms→1.2ms)을 달성하였다.

또한 지식 증류 기반의 학습 전략을 통해 구조 변경에 따른 정확도 손실을 효과적으로 복구하여, 최종적으로 92.89%의 높은 비트 정확도를 유지하였다. 발열 특성 실험에서도 평균 온도 0.67% 감소, 최대 온도 0.27°C 감소를 확인함으로써 열 효율 개선을 실증하였다. 이상의 결과를 바탕으로, 본 연구의 주요 기여는 다음과 같다.

1. 실제 모바일 NPU 병목 현상의 정량적 규명
2. NPU 친화적 아키텍처 최적화 기법 개발
3. 체계적인 Knowledge Distillation 전략 수립
4. 실제 하드웨어 환경에서의 극적인 성능 개선 달성
5. 발열 효율성 개선 효과의 실증적 검증

이러한 통합적 접근은 단순 이론적 모델 경량화를 넘어, 실제 모바일 디바이스에 배포 가능한 실시간 워터마킹 시스템 구현 가능성을 실증한 결과이며, 향후 Edge AI 및 모바일 보안 응용에 널리 활용될 수 있는 기반 기술로 기대된다.

5-2 향후 연구 방향

본 연구는 다음과 같은 한계점을 지닌다. 첫째, ConvTranspose 분해에 따른 구조 변경으로 인해 원본 Lightweight-Mark 모델 대비 약 7%의 비트 정확도 손실이 발생하였다. 이는 일부 민감한 응용 분야에서 성능 보완이 필요한 요소로 작용할 수 있다. 둘째, 제안된 최적화 기법이 발열 특성을 개선함으로써 에너지 효율에 긍정적 영향을 미쳤을 가능성이 있음에도 불구하고, 실제 배터리 소모량이나 전력 소비에 대한 정량적 측정은 수행되지 않았다. 셋째, 본 연구는 주로 CIFAR-10과 128 × 128 크기의 이미지 테스트셋을 사용하여 실험을 진행하였기 때문에, 고해상도 이미지나 동영상과 같은 실환경 데이터에 대한 성능 검증이 추가로 필요하다.

이러한 한계점을 보완하기 위한 향후 연구 방향으로는 다음을 제시할 수 있다. 첫째, 다중 교사 구조(multi-teacher)나 attention 기반 distillation 등 보다 정교한 Knowledge Distillation 기법을 도입하여 95% 이상의 비트 정확도 달성을 목표로 한다. 둘째, 최적화 모델의 실제 NPU 전력 소비량을 측정하여 CPU, GPU 기반 모델과의 에너지 효율성을 비교 분석할 계획이다. 셋째, 본 연구는 NPU 아키텍처 최적화의 가능성을 검증하기 위해 CIFAR-10 기반의 프로토타입 실험을 수행하였으나, 고해상도 이미지나 동영상과 같은 실환경 데이터에 대한 검증이 추가로 요구된다. 향후 연구에서는 DIV2K, COCO 등 고해상도 데이터셋을 적용하여 제안 기법의 실사 이미지 품질과 견고성(Robustness)을 심층적으로 분석할 예정이다.

본 연구는 이론적 모델 경량화와 실제 하드웨어 최적화 간의 간극을 구조적 및 학습적 측면에서 통합적으로 해소하였으며, 모바일 환경에서의 실시간 워터마킹 시스템 구현 가능성을 실제 디바이스 기반의 실험을 통해 실증하였다. 이러한 결과는 향후 모바일 AI 응용 분야에서 실용적 활용 가능성을 제시함과 동시에, 워터마킹 기술의 학문적 발전에도 기여할 수 있을 것으로 기대된다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 ‘메타버스 융합대학원’ 과제(RS-2022-00156318)와, 문화체육관광부 및 한국콘텐츠진흥원의 ‘메타버스에서 저작권 보호 및 이용 활성화 기술 개발’ 과제(RS-2023-00219237)의 지원을 받아 수행되었다.

참고문헌

[1] N. C. Sy, H. H. Kha, and N. M. Hoang, “An Efficient Robust Blind Watermarking Method Based on Convolution Neural Networks in Wavelet Transform Domain,” *International Journal of Machine Learning*, Vol. 10, No. 5, pp. 675-684, 2020. https://doi.org/10.18178/ijmlc.2020.10.5.990

[2] Y. Qiu, H. Fang, and E.-C. Chang, “Lightweight-Mark: Rethinking Deep Learning-Based Watermarking,” in *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, Vancouner: Canada, 2025.

[3] S. Baluja, “Hiding Images in Plain Sight: Deep Steganography,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, Long Beach: CA, pp. 2066-2076, December 2017.

[4] M. Plata and P. Syga, “Robust Spatial-Spread Deep Neural Image Watermarking,” in *Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Guangzhou: China, pp. 62-70, 2020. https://doi.org/10.1109/TrustCom50675.2020.00022

[5] Z. Wang, O. Byrnes, H. Wang, R. Sun, C. Ma, H. Chen, ... and M. Xue, “Data Hiding with Deep Learning: A Survey Unifying Digital Watermarking and Steganography,” *IEEE Transactions on Computational Social Systems*, Vol. 10, No. 6, pp. 2985-2999, 2023. https://doi.org/10.1109/TCSS.2023.3268950

[6] G. Ye, J. Gao, Y. Wang, L. Song, and X. Wei, “ItoV: Efficiently Adapting Deep Learning-Based Image Watermarking to Video Watermarking,” in *Proceedings of the 2023 Internatinal Conference on Culture-Oriented Science and Technology (COST)*, Xi’an: China, pp. 192-197, 2023. https://doi.org/10.1109/COST60524.2023.00047

[7] S. I. Venieris, M. Almeida, R. Lee, and N. D. Lane, “NAWQ-SR: A Hybrid-Precision NPU Engine for Efficient on-Device Super-Resolution,” *IEEE Transactions on Mobile Computing*, Vol. 23, No. 3, pp. 2367-2381, March 2023. https://doi.org/10.1109/TMC.2023.3255822

[8] Qualcomm Technologies, Inc. Qualcomm Hexagon Tensor Processor [Internet]. Available: https://hc2023.hotchips.org/assets/program/conference/day2/ML%20Inference/HC2023%20Qualcomm%20Hexagon%20NPU.pdf.

[9] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and Efficient Object Detection,” in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle: WA, pp. 10778-10787, 2020. https://doi.org/10.1109/CVPR42600.2020.01079

[10] Z. Zhang, P. Zhang, Z. Xu, and Q. Wang, “Reduce Computational Complexity for Convolutional Layers by Skipping Zeros,” arXiv:2306.15951, 2023. https://doi.org/10.48550/arXiv.2306.15951

[11] A. Murtada, O. Abdelrhman, and T. A. Attia, “Mini-ResEmoteNet: Leveraging Knowledge Distillation for Human-Centered Design,” arXiv:2501.18538, 2025, https://doi.org/10.48550/arXiv.2501.18538

김태완(Tae-Wan Kim)

2002년 : 부산대학교 미술학과
2017년 : 가천대학교 게임대학원 게임학 석사
2024년～현 재: 서강대학교 가상융합전문대학원 테크놀로지 전공 박사과정
※관심분야 : 생성형 인공지능(Generative AI), 게임 콘텐츠 개발 자동화(Game Content Automation), 오픈소스 인공지능 모델 응용(Application of Open-source AI Models)

조동헌(Dong-Heon Cho)

2017년 : 연세대학교 정보산업공학, 컴퓨터과학 학사
2019년 : 연세대학교 산업공학 석사
2023년～현 재: 서강대학교 가상융합전문대학원 테크놀로지 전공 박사과정
※관심분야 : 컴퓨터비전(Computer Vision), 최적화(Optimization)

최승관(Seung-kwan Choi)

2000년 : 호서대학교 전자계산학과 석사
2010년 : 세종대학교 디지털콘텐츠학과 박사
2001년～2008년: 한국애니메이션 고등학교 교사
2008년～2021년: 서강대학교 미래교육원 교수
2014년～2016년: 케이크테라피 개발이사
2022년～현 재: 서강대학교 가상융합전문대학원 교수
※관심분야 : 가상융합(Virtual Convergence), XR, SW저작권(Software Copyright)