https://doi.org/10.7840/kics.2025.50.1.1

적응적 발열 관리를 통한 On-Device VSR 성능 최적화

박 혜 진◆, 하 란°

Optimizing On-Device VSR Performance via Adaptive Thermal Management

Hye-jin Park◆, Rhan Ha°

요 약

모바일 기기에서 VSR (Video Super Resolution) 작업은 강한 GPU 연산 부하로 기기 온도를 급격히 상승시킨다. 발열을 해소하기 위해 모바일 시스템에서 보편적으로 사용되는 DVFS (Dynamic Voltage-and-Frequency Scaling)는 애플리케이션 동작 특성을 고려하지 않고 GPU 주파수를 자동으로 감소시켜 급작스럽게 작업의 성능을 저하시킨다. 이러한 열 제한의 발생은 돌발적인 작업 시간 지연을 유발하여 작업 효율성과 사용자 만족도를 저해한다. 본 논문에서는 모바일 VSR에서 발열 문제 해소를 위한 최초의 적응적 발열 관리 전략인 ATM (Adaptive Thermal Management for VSR)을 제안한다. ATM은 기기 온도 변화에 따라 VSR 작업의 추론 단계를 적응적으로 제어하여 온도 증가율을 낮추고 열 제한의 발생 시점을 뒤로 미룬다. 실험을 통해 ATM이 주어진 작업 기간 중 열 제한의 발생을 효과적으로 예방하였으며 모델 추론 속도가 1.56배 저하되는 것을 방지하고 전체 작업 수행량에서 1.8배 향상된 성능을 확인하였다.

키워드 : 초해상화, 발열 관리, 모바일 다비이스, 인터미턴트 컴퓨팅, 머신 러닝
Key Words : super-resolution, thermal management, mobile device, intermittent computing, machine learning

ABSTRACT

VSR (Video Super Resolution) tasks on mobile devices cause a rapid rise in device temperature due to heavy computational load in GPU. Mobile systems commonly employ DVFS (Dynamic Voltage-and-Frequency Scaling) for heat dissipation, but it automatically reduces GPU frequency without considering application behavior, causing a sudden drop in task performance. The occurrence of thermal throttling causes sudden work delays, hindering task efficiency and user satisfaction. Therefore, we propose he first adaptive thermal management technique ATM (Adaptive Thermal Management system for VSR) to address thermal issues. ATM adaptively controls the inference stage of VSR tasks based on device temperature changes to mitigate the rate of temperature increase and delay the onset of thermal throttling. Experiments confirmed that ATM effectively prevents thermal throttling during a given task period, while also preventing a 1.56x decrease in model inference speed and achieving an overall task throughput improvement of 1.8x.

※ 이 논문은 2022학년도 홍익대학교 학술연구진흥비에 의하여 지원되었음
◆ First Author : Hongik University Department of Computer Engineering, hyejin2371@gmail.com 학생회원
° Corresponding Author : Hongik University Department of Computer Engineering, rhanha@hongik.ac.kr, 정회원
논문번호 : 202407-155-A-RN, Received July 24, 2024; Revised August 28, 2024; Accepted September 10, 2024

Ⅰ. 서 론

현대 모바일 기술의 발전과 디바이스 성능의 향상으로 인터넷 공유 플랫폼, 스트리밍 서비스 등 비디오 콘텐츠의 소비가 급증했다. 더불어 사용자에게 더 높은 품질의 미디어 콘텐츠를 효율적으로 제공하기 위한 기술의 필요성이 증대되고 있다. 딥러닝 기술의 발전과 함께 이러한 요구를 충족하기 위한 해결책으로 저해상도 이미지를 고해상도로 복원하는 SR (Super Resolution)이 주목받고 있다. SISR (Single Image Super Resolution)에 이어 VSR (Video Super Resolution)이 최근 몇 년 동안 비디오 콘텐츠 서비스뿐 아니라 의료 영상 처리, 원격 탐사 등 많은 비디오 서비스 분야에서 유용하게 활용되었다. 그중 On-Device VSR은 서버에 연결할 필요 없이 모바일 기기에서 자체적으로 VSR을 수행할 수 있는 기술로, 클라우드 기반 VSR과 달리 네트워크 지연 및 대역폭 제한에 상관없이 VSR 서비스를 제공할 수 있어 활발히 연구되고 있다[1].

VSR은 GPU에서 강도 높은 연산을 연속적으로 수행하기 때문에 모바일 환경에서 장기간 작동 시 기기의 과열 문제가 발생한다. 장치의 수명과 성능을 보호하기 위해서는 발열 해소가 필수적으로 이루어져야 하나 모바일 환경은 공간적 제약으로 인해 활성 냉각 시스템을 사용할 수 없다. 따라서 모바일 기기에는 쿨링팬 등 하드웨어 측면의 발열 관리 대신 소프트웨어 기반의 냉각 시스템인 DTM (Dynamic Thermal Management) 이 주로 사용된다. DTM은 기기 온도를 모니터링하고 제어하기 위한 기술로, 그 중 DVFS (Dynamic Voltage-and-Frequency Scaling)[2]는 프로세서의 전압과 주파수를 동적으로 조정하여 관리하는 기능이다. 기기 온도가 증가하여 임계값에 도달하면 운영 체제가 GPU 주파수를 낮추고 소비 전력을 줄이는 방식으로 온도를 안정화한다. 이러한 열 제한 상태는 기기 온도가 충분히 낮아졌다고 판단할 때까지 유지되고, 이후 다시 최대 주파수를 회복한다.

일반적으로 DVFS는 운영 체제가 제어하기 때문에 발열 관리 시 애플리케이션의 동작 특성을 고려하지 않는다. VSR 작업 중 모바일 시스템이 발열 해소를 위해 GPU 주파수를 급격히 감소시키면 돌발적인 성능 저하가 발생한다. 이는 VSR 작업 시간 지연과 함께 다른 프로세스의 동작에도 악영향을 끼쳐 결과적으로 비디오의 품질 및 사용자 만족도 저하를 일으킨다.

본 연구는 발열로 인한 성능 저하를 해결하고 일정한 성능을 지속적으로 제공하는 ATM (Adaptive Thermal Management for VSR)을 제안한다. ATM은 온도 변화에 적응적으로 작업량을 조절하는 발열 관리 기술로, 효과적인 발열 해소를 통해 VSR 작업의 성능을 최적화한다. 2장에서 모바일 환경에서의 발열과 모바일 VSR에 관한 선행 연구를 분석한다. 3장에서 ATM의 구조와 작동 방식을 설명하고 4장에서 실험 결과를 통해 ATM의 VSR 성능 개선 효과를 평가한다.

Ⅱ. 관련 연구

모바일 환경에서의 발열 대책은 꾸준히 연구되어왔다. 발열 해소 전략은 주로 하드웨어와 소프트웨어 수준의 해결 방법으로 나누어진다. 그러나 하드웨어 측면의 접근 방법은 개발 기간 및 비용이 크게 소모되고 애플리케이션의 요구를 즉각적으로 반영하기 어려워 최근에는 소프트웨어 측면의 해결책이 활발하게 연구되고 있다. DVFS 또한 소프트웨어 측면의 접근 방법 중 하나로 성능과 발열의 trade off를 고려한 발열 관리 기법이다. 그러나 DVFS는 대부분 운영체제 수준에서 관리가 이루어져 신속한 대응이 어렵고 서비스 품질을 고려하지 않아 큰 성능 변동성을 유발한다. 이러한 문제를 해결하기 위해 환경 변화와 어플리케이션 별 리소스 요구사항을 고려하는 DVFS[3] 등이 제안되었다. 하지만 이 방법 또한 어플리케이션의 QoS (Quality of Service)를 고려하지 않고 CPU 및 GPU 주파수를 낮추므로 실제 응용에서 프레임 드롭 등의 문제가 발생할 우려가 있기에 VSR 작업에는 적합하지 않다.

모바일 환경의 기술적 제약을 고려한 효율적인 VSR 기법 또한 다양하게 개발되었다. 모바일 운용을 위한 다양한 저전력, 경량화 SR 모델 구조 개선에 이어, 모바일 VSR의 장기적 성능 향상을 위한 운영 방법도 연구되었다. 에너지 소모가 큰 DNN (Deep Neural Network)모델 작동 시 저전력 모델과의 전환을 적용하여 열 제한의 발생을 방지하거나[4], 열 제한 상태 하에서도 실시간 스트리밍의 QoS를 보장하며 VSR 작업을 완료하기 위해 multi exit 모델 구조를 활용하는 방법이[5] 제시되었다. 하지만 모바일 VSR 기법은 발열 측면이 사전에 고려된 연구가 없고 대부분이 고정적인 VSR을 수행하여 세밀한 작업 제어가 어렵다는 한계가 존재한다.

DVFS는 발열에 즉각적인 대응이 어렵고 VSR 성능에 큰 변동을 야기하며, 기존의 모바일 VSR 기법은 발열 대책 연구와 정밀한 사전 발열 관리 방안이 연구되지 않았다. 본 연구에서 제안하는 ATM은 multi exit 네트워크 구조를 통해 이러한 취약점을 개선한다. Multi exit 구조는 여러 개의 exit point를 통해 중간층에서도 출력

을 생성할 수 있는 네트워크 구조다. 이 구조에 기반한 multi exit 모델은 입력 데이터가 모든 계층의 연산을 거쳐야 하는 기존 DNN 구조와 달리 조건에 따라 일부 연산 과정의 생략이 가능하여 전체적인 연산 부하를 낮추고 추론 시간을 절약할 수 있다. 이러한 장점은 소형 시스템 상의 신경망 추론[6], 실시간 추론 시간 보장과 같이 리소스 및 시간 효율성을 위해서는 활용되었으나 이를 발열 관리에 적용한 연구는 존재하지 않는다. ATM은 최초로 multi exit 구조를 발열 관리에 적용하는 시도이며 기존 VSR이나 발열 관리 방법보다 더욱 유연하고 세밀하게 동작하여 장기간 수행되는 모바일 VSR의 성능을 효과적으로 최적화한다.

Ⅲ. 본 론

ATM은 실시간으로 온도를 측정하고 모델의 동작을 조정하는 ATM 컨트롤러와 단계적으로 중간 결과를 출력할 수 있는 VSR 모델로 구성되며 상황에 맞춰 적응적인 VSR 작업을 수행한다. 이처럼 유연한 작업 수행은 조건에 따라 추론을 조기 종료할 수 있는 multi exit 네트워크 구조를 통해 실현된다. 구체적으로, 컨트롤러는 기기 온도가 임계값에 가까워질수록 모델 추론을 초기 단계에 종료하고 여유가 있다면 추론을 계속 진행하도록 exit을 결정한다. 모델은 exit에 따른 연산을 수행한 후 결과를 출력하여 유동적으로 작업량을 조절한다. 이처럼 온도 변화에 따른 적응적 동작 제어를 통해 기기의 과열을 방지하고 열 제한의 영향을 최소화한다.

3.1 ATM 컨트롤러

ATM 컨트롤러는 모델의 입출력 버퍼를 준비하고 실시간 온도 정보가 VSR 작업에 반영되도록 모델을 실행한다 (그림 1). 각 exit의 경계 온도가 열 제한 임계 온도 값을 기준으로 사전에 설정되고 VSR 작업이 기기 온도와 온도 변화율에 따라 실시간으로 조정된다. 이후 매초마다 GPU 온도를 측정하여 미리 설정된 경계 온도와 비교하고 exit을 결정한다. 온도만을 기준으로 exit을 결정하면 작업 후반부에 낮은 단계의 exit이 빈번하게 선택되어 평균 성능이 저하된다. 따라서 ATM은 온도 증가율이 0 이하인 상황에서는 다음 단계의 exit을 선택하여 온도가 감소하는 추세가 나타나면 작업 강도를 높이도록 동작한다. 이와 같은 방식으로 최대 주파수에서의 작동 시간을 연장하고 높은 성능을 최대한 오래 지속할 수 있도록 한다.

알고리즘 2는 위와 같은 발열 관리 방법을 의사 코드로 나타낸 것이다 (그림 2). ATM은 작업 초기에 미리

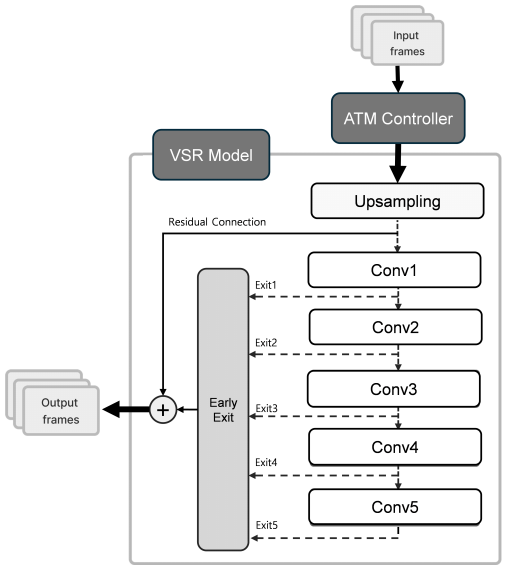


그림 1. ATM 상세 구조
Fig. 1. Architecture of ATM

추정한 임계 온도를 바탕으로 각 exit에 해당하는 경계 온도를 설정한 후 실시간 GPU 온도에 적절한 exit을 선택하여 모델을 실행한다. 입력된 저해상도 프레임은 먼저 2배 크기로 확장되고 exit 번호만큼의 컨볼루션 레이어를 지나며 해상도를 복원한 후 초기 데이터를 합성하여 손실된 잔여 정보를 보완한다.

임계 온도는 기기와 제조사 별로 다르고 정확한 값을 구할 수 없어 exit 구간 설정을 위한 임계 온도는 Android Thermal API[7]의‘Thermal Headroom’을 이용해 추정한다. Thermal headroom은 그동안 측정된 온도 데이터를 기반으로 미래의 온도를 예측한 후 현재 임계 온도까지 남은 여유량을 정규화한 값이다. 그러나 이 값은 기록된 모든 온도 데이터를 계산에 반영하고 갱신된 최댓값만을 반환하기 때문에 기기의 온도 변화를 정확하게 나타내기 어려워 ATM에서 그대로 사용하기에는 적합하지 않다.

ATM은 thermal headroom을 반영하여 새롭게 온도 변화율을 계산하고 임계 온도를 추정한다. ATM에서는 일정 구간의 온도 변화율을 분석하여 온도가 충분히 감소하고 안정적인 상태로 회복되었을 때 exit을 신속하게 전환할 수 있어야 한다. 본 연구의 실험 환경에서 multi exit을 통한 작업량 조절 시 온도가 10~15초 이내에 초기 수준으로 회복되었다. 온도 증가율 측정 구간이 30초 이상으로 너무 길면 최신 온도 변화의 경향성을 정확하게 반영할 수 없으며, 10초 이내로 너무 짧은 경우에는 예외적인 상황에서 민감하게 반응하여 안정적

Algorithm 1 calculateThresholds

```
procedure CALCULATETHRESHOLDS
    temp ← GETTEMPERATURE
    thermalHeadroom ← GETTHERMALHEADROOM
    baseThreshold ← temp − 0.3 × thermalHeadroom + 30
    thresholds ← baseThreshold × [0.8, 0.75, 0.7, 0.65, 0.6]
end procedure
```

Algorithm 2 AdaptiveThermalConrtol

```
procedure ADAPTIVETHERMALCONRTOL
    while true do
        currentTemp ← GETTEMPERATURE
        changeRate ← GETCHANGERATE
        nextExit ← FINDEXIT(currentTemp, thresholds, changeRate)
    end while
end procedure

function FINDEXIT(currentTemp, thresholds, changeRate)
    for i = 0 to 5 do
        if currentTemp < thresholds[i]∧ (changeRate ¡ 0 ∨ i == 5) then
            return exit_i
        end if
    end for
    return exit_1
end function
```

그림 2. ATM 컨트롤러의 exit 결정 알고리즘
Fig. 2. Exit Decision Algorithm of the ATM Controller

인 온도 변화 패턴을 파악하기 어렵다. 이에 따라 온도가 회복되는 시간과 회복 후 안정적인 상태를 유지하는 시간을 고려하여 온도 증가율 계산 구간을 15초로 설정하였다.

결론적으로 온도 변화율은 최근 시점에서 15초 동안의 데이터를 사용하며 식 (3)과 같이 시간(T) 와 온도(t)의 공분산을 시간에 대한 분산으로 나누어 계산한다. 임계 온도는 작업이 시작되기 전의 thermal headroom 값과 온도를 비교하여 추정되고 이를 기반으로 알고리즘 1과 같이 각 exit의 경곗값이 설정된다 (그림 2).

$$V(t) = \sum_{i=1}^{n} \left(\tau_i - \bar{\tau}\right)^2 \tag{1}$$

$$Cov(t) = \sum_{i=1}^{n} \left(t_i - \bar{t}\right)\left(\tau_i - \bar{\tau}\right) \tag{2}$$

$$ChangeRate = \frac{Cov(t,\tau)}{V(\tau)} \tag{3}$$

$t = temperature,\ \tau = time$
$\bar{t} = avreage\ temperature,\ \bar{\tau} = average\ time$

안드로이드 시스템에서 열 제한은 지정된 임계 온도의 85%에서 발생한다. 열 제한에서 회복되기까지 오랜 시간이 걸리고 작업 효율성이 감소하기 때문에 열 제한이 발생하기 전에 충분히 온도를 조절할 수 있도록 임계 온도의 80%를 exit 경계의 최대 온도로 설정한다. 나머지 exit의 경계 온도는 임계 온도의 60%부터 80%까지

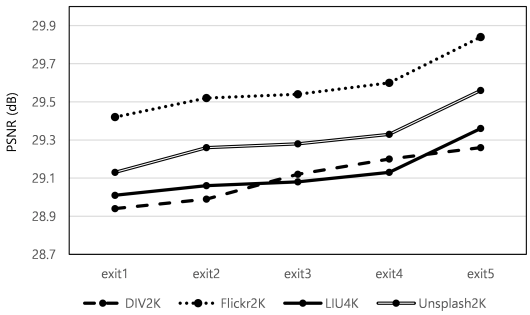


그림 3. 데이터셋 별 exit 에 따른 PSNR 비교
Fig. 3. Comparision of PSNR according to exit across different datasets

를 일정한 간격으로 나눠 결정한다.

3.2 ATM 적용 VSR 모델

ATM에서 사용하는 VSR 모델은 여러 개의 컨볼루션 레이어로 이루어진 모델로, 입력 데이터가 통과하는 레이어의 수가 증가할수록 추론 시간과 결과물의 품질이 함께 증가한다. 레이어가 깊어짐에 따라 안정적으로 품질을 향상시키기 위해 VDSR[8]과 같이 residual connection을 네트워크 구조에 적용한다. Residual connection은 현재 레이어의 연산 결과에 이전 레이어의 연산 결과를 더하여 정보 손실을 방지하는 방법으로, ATM에서는 VSR 모델의 레이어가 깊어져도 residual connection을 적용하여 고해상도의 이미지 출력을 보장한다.

모델의 학습과 성능 평가에는 DIV2K 데이터셋[9]을, 발열 테스트에는 REDS 데이터셋[10]을 사용했고 모든 입력 데이터는 증강된 후 (960, 540) 의 해상도로 전처리 되었다. ATM은 입력된 데이터에 대해 2x super resolution 작업을 수행하며, 출력 결과가 유의미한 품질의 차이를 보이는 5개의 exit을 갖도록 구현된다. 각 exit 별 성능 평가는 Galaxy Note 20에서 DIV2K, Flickr2K[11], Unsplash2K[12], LIU4K[13] 데이터셋 각각 200장을 대상으로 작업한 결과의 평균 PSNR을 산출하여 이루어졌다. 모든 데이터셋에서 품질이 exit을 지날수록 평균 0.2dB씩 점진적으로 향상되었으며, 초기 단계의 exit에서 최소한의 작업만으로 27dB 이상의 양호한 품질을 보장할 수 있음을 확인하였다 (그림 3).

Ⅳ. 성능 평가

실험에서는 ATM과 ATM이 적용되지 않은 VSR의 실행 결과를 비교한다. 후자는 ATM과 같이 컨볼루션 레이어와 residual connection을 사용하는 EVSRNet[14]

을 사용한다. ATM은 모바일 VSR에서 발열 문제 해결을 위한 최초의 시도로, 동일한 목적 및 접근 방법의 연구가 이전에는 이루어지지 않아 기존 방법과 직접적인 비교가 불가능하다. 따라서 성능 평가는 같은 작업에 대한 ATM 수행 결과와 기본 안드로이드 시스템에서의 VSR 수행 결과를 비교하여 이루어졌다.

실험은 Adreno 650 GPU를 탑재한 Galaxy Note 20 상에서 진행되었으며 모든 경우에 기기를 약 40도까지 냉각시킨 후 수행되었다. REDS 데이터셋을 사용하여 30분 동안 VSR 작업을 수행하고, 총 20회 반복하여 일관된 결과를 확인했다. 작업 중 기기의 GPU 온도 변화, 열 제한 발생 상태, 프레임 당 추론 시간, GPU 주파수가 매초마다 기록되었고, GPU 온도와 주파수는 내장된 센서를, 열 제한 상태는 Android Thermal API의 thermal status를 사용해 측정했다.

관찰된 성능 지표를 분석하여 다음과 같이 ATM의 기능을 평가하고자 한다. 첫째, VSR 작업 중 온도 변화와 그에 따른 열 제한 발생 상태를 관찰하여 ATM의 발열 제어 능력을 평가한다. 둘째, 열 제한 발생 전후의 모델 추론 시간을 비교하여 ATM의 발열 관리가 VSR 작업 시간을 얼마나 효과적으로 개선할 수 있는지 분석한다. 마지막으로 모델 추론을 포함한 전체 작업의 수행량을 비교하여 작업 효율성 향상을 확인한다.

4.1 열 제한 발생 지연 효과

발열 관리를 수행하지 않는 VSR은 고강도 연산을 초기에 집중해서 수행한다. 반면 ATM은 작업량을 조절하여 고강도 연산을 작업 전반에 나누어 진행하게 되고 높은 GPU 주파수와 낮은 GPU 주파수 상태를 분산시킨다. 이는 급격한 온도 상승을 방지하고 온도를 점진적으로 높이게 되어 임계값에 도달하는 시점을 지연시킨다. 그림 4의 (a), (b)는 작업 중 임의의 구간에서의 GPU 주파수 변화를 보여주며 ATM 수행 여부에 따른 차이를 나타낸다.

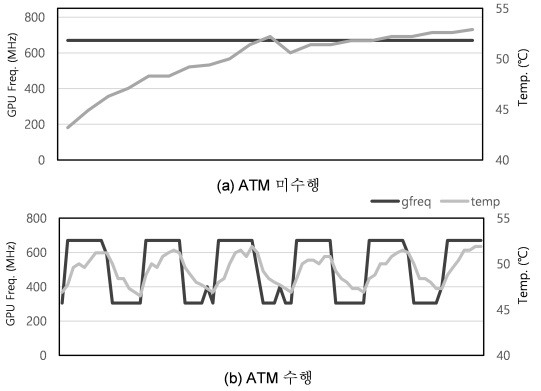


그림 4. ATM 수행 여부에 따른 GPU 주파수 변동 비교
Fig. 4. Comparison of GPU Frequency depends on ATM

ATM을 적용하지 않은 그림 4(a)에서는 초기에 GPU 주파수를 최대 상태로 유지하여 GPU 온도가 계속해서 상승한다. ATM의 수행 결과인 그림 4(b)에서는 추론 강도가 유동적으로 조절되어 GPU 주파수가 빈번

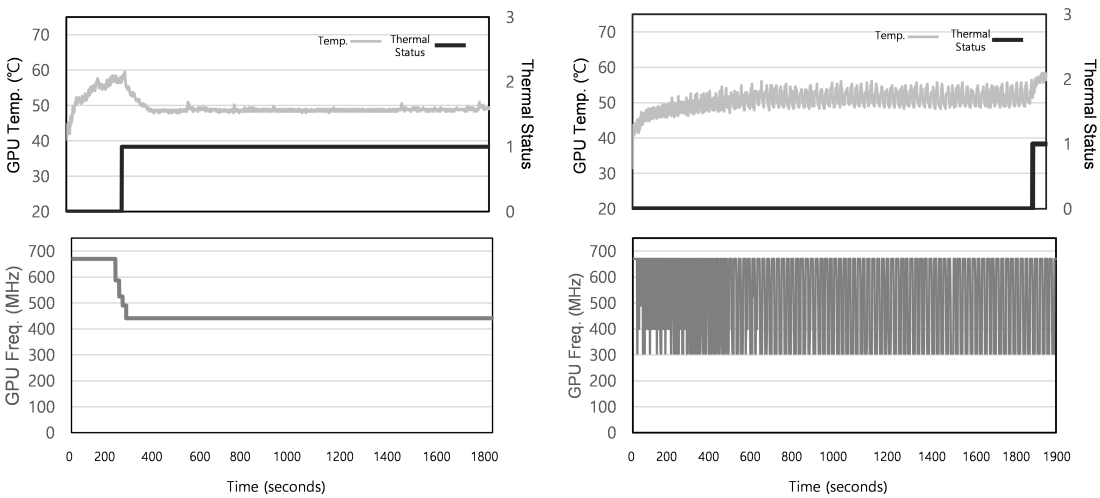


그림 5. ATM 수행 여부에 따른 GPU 온도, 열 제한 상태, GPU 주파수 변화
Fig. 5. Traces of GPU Temperature., Thermal Throttling Status, and GPU Frequency depends on ATM

하게 변동하는 패턴을 반복한다. 높은 GPU 주파수와 낮은 GPU 주파수가 전환되면서 작업 전반에 걸쳐 연산 부하가 분산되고 온도 증가율이 감소하여 열 제한 발생 시점이 뒤로 미루어진다. 이와 함께 작업 초기에만 최대 GPU 주파수를 유지하는 4(a)와 달리 4(b)는 최대 GPU 주파수를 시간적으로 분산시켜 더 오랜 기간 열 제한 발생 없이 사용할 수 있다.

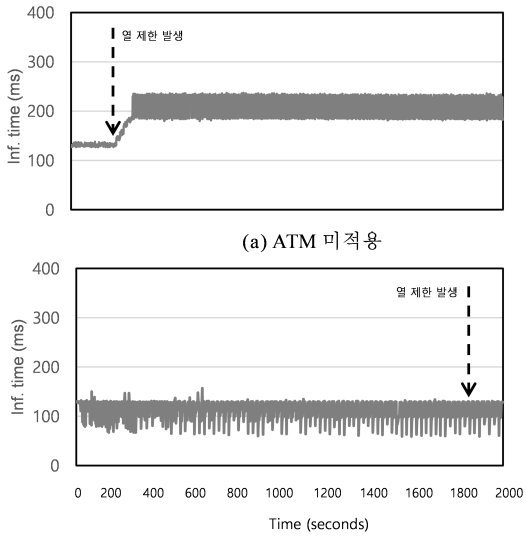
다음으로 ATM 수행 여부에 따른 온도 변화와 열 제한 발생 여부를 비교한다. ATM을 적용하지 않은 VSR의 경우 작업 초기에 기기가 고강도 연산을 연속적으로 처리하기 위하여 최대 GPU 주파수를 유지하여 GPU 온도가 빠르게 상승한다. 그림 5(a), (b)는 ATM 수행 여부에 따른 GPU 온도, 열 제한 상태, GPU 주파수의 변화를 보여준다. ATM이 적용되지 않은 5(a)에서는 작업을 시작한 지 300초 후에 열 제한이 발생하여 GPU 주파수가 670MHz에서 440MHz로 낮아진다.

ATM의 수행 결과에서는 앞서 언급한 연산 부하 분산 효과로 GPU 온도가 서서히 상승한다. 그림 5(b)에서 보여지듯이 GPU 주파수는 급격히 낮아지지 않고 빈번하게 변동하며 높은 GPU 주파수 작업을 고르게 분산시켜 전반적인 추론 성능을 유지한다. VSR 작업을 30분간 진행한 결과 ATM 수행 시에는 작업이 종료되는 시점까지 열 제한이 발생되지 않았다. 추가로 시간을 늘려 40분 동안 ATM을 수행한 결과에서 31.6분 시점에 열 제한이 발생되는 것을 확인하였다. 결과적으로 ATM을 미적용한 VSR 작업에서는 작업 초기에 열 제한이 발생되었으나 ATM 수행 시에는 30분의 실험 동안 열 제한을 완전히 방지했다.

4.2 모델 추론 성능 개선 효과

본 절에서는 GPU 주파수와 그에 따른 추론 시간 변화를 비교하여 열 제한 방지를 통한 ATM 시스템의 추론 성능 개선 효과를 확인한다 (그림 6). ATM은 유동적인 발열 관리를 통해 열 제한 발생 시점을 늦추고 추론 성능을 더 오래 유지한다. 더불어 열 제한이 발생하더라도 온도 변화에 적응적으로 연산 강도를 변경하여 안정적인 모델 추론 속도를 제공한다.

그림 6은 열 제한으로 인한 추론 시간 변화의 차이를 ATM 적용 여부에 따라 비교한 것이다. ATM을 적용하지 않는 그림 6(a)에서는 작업 시작 후 300초부터 열 제한이 발생하여 모델의 추론 속도가 평균 134ms/frame에서 210ms/frame으로 1.56배 낮아진다. 반면 ATM은 그림 5(b)와 같이 임계 온도 도달 시점이 늦추어지게 되어 작업을 마칠 때까지 일정한 추론 속도를 유지한다. 더불어 ATM은 multi exit 구조를 통해



(a) ATM 미적용

(b) ATM 수행 시

그림 6. ATM 수행 여부에 따른 추론 시간 변화
Fig. 6. Traces of inference time depends on ATM

열 제한으로 인한 성능 저하 상황에서도 서비스 품질을 일정 수준 보장한다. 그림 6(b)에서 ATM은 열 제한이 발생된 후에도 낮은 단계의 exit을 선택하여 PSNR 측면의 품질을 허용 범위 내에서 일부 포기하더라도 추론 속도를 거의 초기와 유사한 수준인 132ms/frame으로 계속 유지한다.

결론적으로 ATM은 열 제한 발생을 방지할 뿐 아니라 열 제한이 발생하더라도 적응적인 제어를 통해 모델의 추론 성능을 안정적으로 유지하며, 이는 장기적인 작업 성능 향상에도 기여한다. 4가지 데이터셋을 대상으로 30분 동안 ATM을 수행한 결과, 기존 안드로이드 시스템에 비해 추론 속도가 최소 1.43배에서 최대 1.76배 향상되었다. ATM 수행에 따른 평균 PSNR 저하는

표 1. 데이터셋 별 30분간 ATM 수행 여부에 따른 성능 비교
Table 1. Comparison of performance depedns on ATM over 30 minutes across different datasets.

Dataset	Average Inf. time (sec / frame)		Average PSNR (dB)	
	ATM 미적용	ATM 수행	ATM 미적용	ATM 수행
REDS	0.189	0.132	29.01	29.27
Flickr2K	0.23	0.134	29.58	29.78
Unsplash 2K	0.197	0.112	28.75	29.57
LIU4K	0.211	0.124	31.09	29.34

약 0.27dB로 미세한 품질 감소가 발생했으며, 이는 ATM에서 작업 시간과 품질의 trade off가 합리적임을 나타낸다 (표 1).

4.3 작업 수행 성능 개선 효과

사용자 만족도에 직접적인 영향을 미치는 요소는 전체 프로그램의 수행 시간이다. 따라서 ATM의 모델 추론을 포함한 전체 작업 성능 개선 효과를 추가 분석하기 위해 VSR 모델 추론에 결과 이미지 변환 작업을 포함한 총 작업 수행 시간의 변화를 관찰했다. 30분 동안 작업을 수행한 결과 ATM 미적용 시에는 약 2,886프레임을 처리했고, ATM 수행 시에는 5,399프레임을 처리했다. 이는 약 1.8배 향상된 결과로, 열 제한 발생과 모델 추론 시간 지연을 예방함으로써 전체 작업 처리량이 향상되었다 (표 2).

결론적으로, 실험을 통해 VSR을 위한 애플리케이션 수준의 발열 관리 시스템인 ATM의 성능 개선을 확인했다. ATM은 적응적인 VSR 추론 제어를 통해 발열을 유발하는 연산 부하를 작업 전반에 분산시키고 온도 상승률을 낮춘다. 그 결과 기기가 임계 온도에 도달하는 기간이 늦추어지고 열 제한의 영향을 최소화하여 모델 추론 시간이 1.56배 증가하는 상황을 예방했다. 모델 추론을 포함한 전체 작업 수행 측면에서는 ATM 미적용 시보다 1.8배 많은 프레임을 처리하여 작업 효율성을 향상하였다.

표 2. ATM 수행 여부에 따른 30분간 작업량 비교
Table 2. Comparison of 30-minute workload depends on ATM

	30분 동안 처리한 프레임 수
ATM 미적용 시	2,886
ATM 수행 시	5,399

V. 결 론

열 제한 발생으로 인한 GPU 주파수 감소는 추론 시간의 증가와 더불어 VSR 작업에 예기치 않은 성능 저하를 초래한다. 모바일 VSR에서 돌발적인 처리 시간 지연은 사용자 만족도를 크게 해칠 수 있어, 추론 시간의 안정성과 지속적인 성능 보장은 매우 중요하다.

본 논문에서 제안된 ATM은 모바일 VSR에서의 열 제한 발생을 방지하고 성능 안정성을 확보하기 위한 적응적 발열 관리 기술로, 유동적인 연산이 가능한 multi exit 신경망 구조를 통해 온도 변화에 적응적인 VSR을 수행한다. ATM은 정교한 사전 발열 관리를 통해 열 제한 도달 시기를 지연시키고 안정적인 성능을 지속하여 장기적인 VSR 수행에서 작업 효율성을 개선한다. ATM은 실험 중 열 제한을 완전히 방지하였으며, 추론 속도가 1.56배 저하되는 문제를 예방하고 전체 작업량을 1.8배 향상시켰다.

ATM은 효과적으로 열 제한을 방지하고 VSR 수행 능력을 개선하여 지속적인 성능과 효율성이 중요한 실시간 스트리밍 또는 대용량 데이터의 On-Device VSR 작업에서 유용하게 활용될 수 있을 것이다. 그리고 현재 사용된 임계 온도의 추정값 대신, 실제 임계 온도와 기기 발열 시스템을 기반으로 더 정밀화한다면 보다 효과적인 발열 관리가 가능할 것으로 예상된다. 추가적으로, 본 논문의 단일 기기에서 연산 부하들을 시간적으로 분산시키는 ATM 기법에 더하여, 작업들을 여러 각 기기의 상황을 고려하여 적절하게 분배할 수 있다면 보다 정교한 성능 개선을 기대할 수 있을 것이다.

References

[1] A. Papanai, S. Babbar, A. Pandey, H. Kathuria, A. K. Sharma, and N. Gupta, “VIhanceD: Efficient video super resolution for edge devices,” in *2023 2nd Edition of IEEE Delhi Section Flagship Conf. (DELCON)*, pp. 1-6, Rajpura, India, Feb. 2023. (https://doi.org/10.1109/DELCON57910.2023.10127275)

[2] S. Lee and Y. Park, “A kernel-aware runtime DVFS system on mobile GPUs,” in *Korea Comput. Congress 2020*, pp. 1216-1218, Online, Jul. 2020.

[3] S. Kim, K. Bin, S. Ha, K. Lee, and S. Chong, “zTT: Learning-based DVFS with zero thermal throttling for mobile devices” in *Proc. 19th Annual Int. Conf. Mobile Syst., Appl., and Services (MobiSys)*, pp. 41-53, NY, United States, Jul. 2021. (https://doi.org/10.1145/3458864.3468161)

[4] Y. Zhou, F. Liang, T. Chin, and D. Marculescu, “Play it cool: Dynamic shifting prevents thermal throttling,” in *ICML Dynamic Neural Netw. (DyNN) Wkshp. 2022*, Baltimore, MD, USA, Jul. 2022. (https://doi.org/10.48550/arxiv.2206.10849)

[5] S. Park, Y. Cho, H. Jun, J. Lee, and H. Cha,

"OmniLive: Super-resolution enhanced 360° video live streaming for mobile devices," in *Proc. 21st Annu. Int. Conf. MobiSys*, pp. 261-274, Helsinki, Finland, Jun. 2023. (https://doi.org/10.1145/3581791.3596851)

[6] Y. Li, Y. Wu, X. Zhang, J. Hu, and I. Lee, "Energy-aware adaptive multi-exit neural network inference implementation for a millimeter-scale sensing system," in *IEEE Trans. VLSI Syst.*, vol. 30, no. 7, pp. 849-859, Jul. 2022. (https://doi.org/10.1109/TVLSI.2022.3171308)

[7] Google, *Thermal mitigate*(2024), android open source project, Retrieved Jul., 2, 2024, from https://source.android.com/docs/core/power/thermal-mitigation#codes.

[8] J. Kim, J. Lee, and K. Lee, "Accurate image super-resolution using very deep convolutional networks," in *2016 IEEE/CVF Conf. CVPR*, pp. 1646-1654, Las Vegas, NV, USA, Jun. 2016. (https://doi.org/10.48550/arXiv.1511.04587)

[9] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *The IEEE/CVF Conf. CVPRW*, pp. 1122-1131, Honolulu, Hi, USA, Jul. 2017. (https://doi.org/10.1109/CVPRW.2017.150)

[10] S. Nah, S. Baik, S. Hong, G. Moon, S. Son, R. Timofte, and K. Lee, "Ntire 2019 challenge on video deblurring and super resolution: Dataset and study," in *Proc. IEEE/CVF Conf. CVPRW*, pp. 1996-2005, Long Beach, CA, USA, Jun. 2019. (https://doi.org/10.1109/CVPRW.2019.00251)

[11] B. Lim, S. Son, H. Kim, S. Nah, and K. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. CVPRW*, pp. 114-125, Honolulu, HI, USA, Jul. 2017. (https://doi.org/10.48550/arXiv.1707.02921)

[12] Y. Kim and D. Son, "Noise conditional flow model for learning the super-resolution," in *Proc. IEEE/CVF Conf. CVPRW*, pp. 424-432, Nashville, TN, USA, Jun. 2021. (https://doi.org/10.48550/arXiv.2106.04428)

[13] J. Liu, D. Liu, W. Yang, S. Xia, X. Zhang, and Y. Dai, "A comprehensive benchmark for single image compression artifact reduction," in *IEEE Trans. Image Process.*, vol. 29, pp. 7845-7860, Jul. 2020. (https://doi.org/10.48550/arXiv.1909.03647)

[14] S. Liu, C. Zheng, K. Lu, S. Gao, N. Wang, B. Wang, D. Zhang, X. Zhang, and T. Xu, "EVSRNet: Efficient video super-resolution with neural architecture search," in *2021 IEEE/CVF Conf. CVPRW*, pp. 2480-2485, Nashville, TN, USA, Jun. 2021. (https://doi.org/10.1109/CVPRW53098.2021.00281)

박 혜 진 (Hye-jin Park)

2025년 2월 : 홍익대학교 컴퓨터공학과 졸업 예정

<관심분야> 기계학습, 온디바이스 컴퓨팅, 뉴럴 네트워크

[ORCID:0009-0006-3000-8638]

하 란 (Rhan Ha)

1987년 : 서울대학교 컴퓨터공학 학사

1989년 : 서울대학교 컴퓨터공학 석사

1989년~1990년 : KT 전임연구원

1995년 : University of Illinois at Urbana-Champaign 컴퓨터공학 박사

1995년~현재 : 홍익대학교 컴퓨터공학과 교수

<관심분야> 머신 러닝, 온디바이스 컴퓨팅, 사물인터넷, 인터미턴트 컴퓨팅, 임베디드 시스템, 모바일 컴퓨팅, 실시간 시스템, SW보안

[ORCID:0000-0001-9861-362X]