Journal of KIIT. Vol. 24, No. 2, pp. 103-112, Feb. 28, 2026. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2026.24.2.103

멀티모달 데이터 처리 환경에서 CPU-GPU 효율성을 고려한 동적 자원 분산 시스템

주유진*[1], 곽성신*[2], 박소현**

Dynamic Resource Distribution System Considering CPU-GPU Efficiency in Multimodal Data Processing Environments

Yujin Ju*[1], Sungshin Kwak*[2], and Sohyun Park**

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 디지털분야글로벌연구지원 연구결과로 수행되었음 (RS-2024-00428758)

요 약

대규모 멀티모달 파이프라인에서 CPU와 GPU 자원 배분에 대한 정량적 기준은 아직 명확히 정립되지 않았다. 본 연구는 GPU가 CPU 대비 속도 2x 이상, 샘플당 에너지 소비 50% 이내일 때만 GPU를 사용하는 결합 임계치를 정의하고, 이에 따라 CPU/GPU를 동적으로 선택하는 경량 DNN 기반 자원 분산 시스템을 제안한다. 이미지·비디오·텍스트·수치 데이터와 다양한 모델 조합에 대해 동일한 프로토콜로 학습 로그를 수집하고, FLOPs, 파라미터 수, 배치 크기, 입력 크기, 하드웨어 지표를 특징으로 실시간 추론 가능한 분류기를 학습하였다. 실험 결과, 소규모 과제에서는 CPU가, 대규모 시각·시계열 과제에서는 GPU가 우수했으며, 제안한 임계치 기반 결정은 불필요한 GPU 사용을 줄이면서 처리량과 에너지 효율을 동시에 개선하였다. 본 시스템은 전처리-학습 전 단계에 적용 가능하며, 강화학습 기반 스케줄러로의 확장 가능성을 보인다.

Abstract

Quantitative criteria for deciding when and how to allocate CPU and GPU resources in large-scale multimodal pipelines remain insufficiently established. This work defines a joint threshold—using the GPU only when it achieves at least a 2× speedup over the CPU while consuming no more than 50% of the per-sample energy—and proposes a lightweight DNN-based resource allocation system that dynamically selects between CPU and GPU based on this criterion. Using a unified protocol, we collect training logs across diverse datasets and representative model combinations spanning image, video, text, and numerical data, and train a lightweight classifier capable of real-time inference using features such as FLOPs, parameter count, batch size, input size, and hardware metrics. Experimental results show that CPUs are preferable for small-scale or lightweight tasks, whereas GPUs dominate large-scale visual and temporal workloads. The proposed threshold-based decision strategy reduces unnecessary GPU usage while simultaneously improving throughput and energy efficiency. The system is applicable to preprocessing and pre-training pipeline stages and demonstrates potential for extension to reinforcement learning-based schedulers.

Keywords

multimodal pipeline, dynamic CPU-GPU resource allocation, joint threshold, speed and energy, improved energy efficiency

* 단국대학교 죽전캠퍼스 인공지능융합학과
- ORCID[1]: http://orcid.org/0009-0001-2198-2061
- ORCID[2]: http://orcid.org/0009-0007-6789-9590
** 단국대학교 죽전캠퍼스 소프트웨어학과(교신저자)
- ORCID: http://orcid.org/0000-0003-2843-9296

· Received: Nov. 10, 2025, Revised: Dec. 31, 2025, Accepted: Jan. 03, 2026

· Corresponding Author: Sohyun Park
Division of Software Convergence, Room 426, Dankook University, 152 Jukjeon-ro, Suji-gu, Yongin-si, Gyeonggi-do 16890, Republic of Korea
Tel.: +82-31-8021-8444, Email: sohyunpark@dankook.ac.kr

Ⅰ. 서 론

대규모 멀티모달 데이터 처리는 최근 인공지능(AI) 연구에서 중요한 과제로 부상하고 있다[1]. 그러나 이미지·텍스트·비디오·수치 등 이질적 데이터를 통합하는 과정은 막대한 연산과 전력을 요구하고, 이는 클라우드 비용과 환경적 부담으로 직결된다[2][3]. 그럼에도 실무에서는 어떤 조건에서 GPU가 '가치 있는 가속'인지, 반대로 CPU로 처리해도 손해가 없는 경우가 무엇인지를 정량적 기준으로 판단하는 체계가 충분히 정립되어 있지 않다. 예를 들어 최신 대규모 멀티모달 모델의 학습·추론은 다수 GPU의 장시간 사용을 요구하여 에너지와 비용 부담을 증가시킨다. 따라서 CPU와 GPU의 상보적 특성을 고려한 효율적 자원 관리 시스템은 단순 성능 향상을 넘어 비용 절감과 지속 가능한 AI를 위해 필수적이다[4][5]. 한편 기존 연구는 정렬·융합 등 모델 성능 향상에 집중되어 왔으며, 데이터 규모·복잡성 변화에 따라 CPU/GPU를 언제·어떻게 배분해야 효율적인지에 대한 체계적 분석과 운영 기준은 부족하다. 최근에는 GPU 자원 공유 및 에너지-aware 스케줄링 연구가 진행되고 있으나[6], 이러한 접근은 GPU 사용 여부 자체보다는 GPU 활용 시 효율 향상에 초점을 둔다. 따라서 CPU와 GPU 중 어느 장치를 선택해야 하는지를 정량적으로 정의하는 기준 설정은 여전히 해결되지 않은 연구 과제로 남아 있다.

이에 본 연구는 멀티모달 파이프라인에서 입력·모델·하드웨어 특성을 바탕으로 CPU와 GPU를 교차 활용하는 자원 분산 시스템을 제안하고, 속도와 에너지의 결합 기준을 사용한 전환 임계를 도입해 실증적으로 검증한다. GPU 사용 여부는 속도와 에너지를 함께 본 결합 기준으로 결정한다. 구체적으로, GPU가 CPU보다 두 배 이상 빠르지 않으면 실무적으로 의미 있는 가속으로 보지 않으며, 샘플 당 에너지 소비가 CPU 대비 50% 이상 불리하다면 명백한 비효율로 간주한다. 이 두 임계는 암달의 법칙[7]과 초기화·데이터 전송 등 부가 오버헤드를 반영해 설정한 보수적 하한으로, 성능/전력 지표 활용 관행을 참조해 정하였다. 최종적으로는 이 속도×에너지 결합 의사결정에 따라 CPU와 GPU를 동적으로 선택한다. 본 연구는 기존 멀티모달 학습 연구들이 모델 성능 향상에 집중하는 동안 학습/추론 과정에서의 자원 활용 기준이 충분히 정립되지 않았다는 한계를 문제로 인식한다. 특히 GPU 가속이 무조건적인 이득으로 간주되는 관행은 입력 규모·연산량·전력 조건에 따른 비용·효율 편차를 간과한다. 따라서 본 연구의 목표는 언제 CPU/GPU를 선택해야 하는지를 정량적으로 정의하고 이를 다양한 데이터·모델·하드웨어 조합에서 실험적으로 검증하는 것이다. 이러한 접근은 멀티모달 통합 효율성을 위한 자원 선택 기준을 명확히 제시하며 이에 대한 실증적 검증이 본 논문의 주요 기여이다.

이 시스템은 이미지·텍스트·비디오·수치 등 서로 다른 특성을 가진 데이터셋을 활용해 CPU와 GPU의 효율을 비교하고, 해당 로그를 입력으로 하는 머신러닝 기반 자원 선택 모델을 학습한다. 구체적으로 Kinetics-400, ImageNet, IMDB, Iris 등 다양한 데이터 - 모델 조합에 대해 CPU/GPU에서 동일 학습을 반복 측정하고, 시간·평균 전력·메모리·utilization 등의 하드웨어 로그와 FLOPs·파라미터·배치·입력 크기 등의 모델/데이터 특징을 수집해 자원 선택 DNN의 입력으로 사용한다[8][9]. 각 조합은 표준화된 프로토콜로 실행되며, 100회 이상 반복의 평균치를 사용해 변동성을 낮췄다.

실험 결과, 작업 규모가 작을 때 초기화 지연과 데이터 전송 오버헤드로 인해 CPU가 효율적임을 확인하였다. 특히 입력 크기·FLOPs·배치 크기가 전환 임계를 지배하며, 제안한 전환 임계에 따라 의사결정 함으로써 불필요한 GPU 사용을 줄이고 전체 처리량과 에너지 효율을 동시에 개선할 수 있었다.

Ⅱ. 관련 연구

2.1 멀티모달 학습과 융합 기법

멀티모달 학습은 이미지, 텍스트, 비디오 등 서로 다른 모달리티를 결합하여 풍부한 표현을 학습하는 것을 목표로 한다. 기존 연구들은 주로 정렬과 융합 기법을 중심으로 발전해 왔다. 예를 들어, 감정 인

식, 의료 영상 진단, 영상 이해 등 다양한 응용 분야에서 멀티모달 융합 모델이 제안되었으며 이미지 캡셔닝과 시각 질의응답과 같은 과제에서 성능 향상이 입증되었다. 그러나 이러한 연구들은 대부분 알고리즘적 성능 개선에 집중하였고 실제 학습 과정에서의 연산 자원 효율성은 충분히 다루지 못했다. 특히 데이터 크기와 복잡성이 달라질 때 CPU와 GPU를 어떻게 효율적으로 활용할 수 있는지에 대한 분석은 부족하다.

2.2 자원 효율성과 분산 시스템 연구

자원 효율성을 다룬 연구들은 주로 GPU 기반 병렬 연산이나 이기종 환경에서의 부하 분산에 초점을 맞추었다. 기존 연구에서는 TPU, GPU, CPU의 연산 성능을 벤치마킹하여 GPU의 우수성을 입증하였으나 CPU의 기여는 제한적으로만 평가하였다. 또 다른 연구에서는 이기종 시스템에서 동적 부하 분산 기법을 제안했으나 GPU 클러스터 관리에 머물러 CPU와 GPU를 상호 보완적으로 활용하는 시스템까지는 확장되지 못했다. 본 연구는 이러한 한계를 해결하기 위해, 다양한 멀티모달 데이터셋을 활용하여 CPU와 GPU의 효율성을 실증적으로 비교·분석하고 머신러닝 기반의 자원 선택 모델을 제안한다.

III. 연구 방법

3.1 데이터셋

본 연구는 다양한 데이터 유형과 규모에 따른 CPU와 GPU의 효율성을 검증하기 위해 표 1과 같은 비디오, 텍스트, 수치, 이미지 데이터셋을 활용하였다. Kinetics-400[10]은 대규모 인간 행동 비디오 데이터셋으로 연산량이 큰 시계열 처리의 특성을 반영한다. IMDB[11] 리뷰 데이터셋은 영화 리뷰와 긍·부정 레이블로 구성된 텍스트 데이터셋으로 비교적 연산량이 적은 과제를 제공한다. Iris[12] 데이터셋은 수치 데이터로 단순 분류 작업의 기준선 역할을 한다. ImageNet[13]은 대규모 범주형 이미지 데이터셋으로 단순 이미지 분류를 가능하게 한다. 각 데이터셋은 공개 벤치마크 데이터셋이며 표준 연구에서 널리 사용되는 대표 과제 유형을 반영하기 위해 선택하였다. Kinetics-400, ImageNet, IMDB 리뷰 데이터셋, IRIS 데이터셋은 각각 시계열 영상, 범주형 이미지, 텍스트 감정분석, 단순 분류의 대표적 벤치마크로 활용되며 이 조합을 통해 입력 규모·연산량에 따른 CPU/GPU 효율 차이를 체계적으로 비교할 수 있다. 이러한 데이터셋 조합은 데이터 유형과 복잡성 변화에 따른 자원 효율성 비교를 가능하게 한다. 각 데이터셋은 개별적으로는 단일 모달리티에 해당하지만 서로 다른 연산 특성과 규모를 가지는 네 가지 작업을 조합함으로써 실제 멀티모달 파이프라인에서 등장하는 이질적 작업들의 혼합을 근사하도록 설계하였다. 따라서 본 실험은 멀티모달 표현 학습 성능 자체보다는 다양한 모달·모델 조합이 공존하는 환경에서 CPU-GPU 자원 선택 기준을 정량화하는 데 초점을 둔다.

표 1. 사용된 데이터셋
Table 1. used datasets

Dataset	Model
Kinetics-400	Timsformer
IMDB	Bert
IRIS	XGB
Imagenet	ResNet-18, MobileNetV3-Small

또한, 각 데이터셋과 모델 조합에 따른 학습 로그 값을 수집하여 본 연구의 시스템 학습에 활용할 최종 데이터셋을 구축하였다. 이 과정에서 데이터셋의 결과값은 다음의 식 (1)을 사용하여 산출하였다. 식 (1)은 시간/에너지 값을 직접 산출하여 회귀적으로 예측하기 위한 식이 아닌 실측 로그 기반으로 CPU 또는 GPU 중 어느 장치를 선택해야 하는지를 결정하는 이진 라벨을 생성하기 위한 기준식이다. 본 연구에서 정답은 식 (1)의 조건을 만족하는 경우 GPU(1), 만족하지 않는 경우 CPU(0)로 부여되는 장치 선택 레이블을 의미한다. 즉, 식 (1)은 예측 대상이 아니라 레이블을 얼마나 잘 근사하는지를 나타내는 분류 정확도이다.

$$\left(\frac{T_{CPU}}{T_{GPU}} < T_t\right) \vee \left(\frac{E_{GPU}/N}{E_{CPU}/N} > T_e\right) \quad (1)$$

본 연구에서는 속도 임계치 $T_t = 2.0$, 에너지 임계치 $T_e = 1.5$로 설정하였다. TDP_{GPU}와 TDP_{CPU}는 각각 동일한 학습 설정에서 CPU 및 GPU로 수행한 학습 소요 시간을 의미한다. E_{CPU}와 E_{GPU}는 학습 구간 평균 전력과 시간을 곱하여 산출한 총 에너지 소비량으로 정의되며, 본 연구에서는 GPU 학습 시 CPU와 GPU가 동시에 사용되는 점을 고려하여 식 (2)와 같이 계산하였다.

$$\begin{cases} E_{CPU} = \overline{P}_{CPU} \cdot T_{CPU} \\ E_{GPU} = (\overline{P}_{CPU} + \overline{P}_{GPU}) \cdot T_{GPU} \end{cases} \quad (2)$$

임계치 T_t는 속도 향상 판단 기준으로, 본 연구에서는 GPU가 CPU 대비 유효 속도 향상이 2배 이상일 때만 실질적으로 가속으로 간주하였다 ($T_t = 2.0$). 이는 암달의 법칙에 따르면 전체 연산 중 직렬 구간의 비율이 일정 수준 존재할 경우, 병렬화 가능한 부분을 아무리 확장하더라도 전체 속도 향상에는 상한이 존재하기 때문이다. 예를 들어 병렬화 가능 비율이 70%일 경우 이론적으로 달성 가능한 최대 속도 향상은 약 3.3배에 불과하며, 여기에 GPU 초기화, 데이터 전송 등 실무 환경에서의 부가적 오버헤드를 고려하면 실제 체감 성능 향상은 더욱 제한된다. 따라서 본 연구에서는 GPU 활용이 명확한 이점을 제공한다고 판단할 수 있는 최소 기준으로 $T_t = 2.0$을 설정하였으며, 식 (1)에 따라 GPU가 해당 임계치를 만족하지 않는 경우에는 CPU로 처리하더라도 서비스 수행에 실질적인 문제가 없다고 판단하였다.

마찬가지로 T_e는 에너지 효율 기준으로, GPU가 CPU보다 샘플당 에너지 소모에서 50% 이상 불리하다고 정의하였다. HPC 분야에서는 Green500 등에서 성능/전력 지표를 공식적으로 활용하여 시스템 효율을 평가하고 있다. 본 연구에서는 이러한 평가 방식에 근거하여, 소폭의 차이는 환경적 변동으로 간주하고 CPU 대비 GPU가 에너지 효율에서 50% 이상 불리한 경우를 명백한 비효율로 판단하였다. 이 수치는 Green500의 공식 임계값은 아니며, 본 연구가 실험적 타당성을 확보하기 위해 설정한 보수적 기준이다.

각 데이터-모델 조합에 대해 동일한 학습 과정을 CPU와 GPU 환경에서 각각 반복 수행하여, 표 2와 같이 보여주고 있다.

표 2. CPU/GPU 선택 라벨 생성을 위한 실측 결과
Table 2. Experimental results for generating CPU/GPU selection labels

Method	Device	Avg_cpu_power	Avg_gpu_power	Elapsed time
bert_imdb	CPU	37.763	-	2138.442
	GPU	37.138	83.993	50.241
mobilenet_imagenet100	CPU	43.998	-	46.396
	GPU	46.104	19.520	25.986
resnet18_imagenet100	CPU	38.718	-	253.113
	GPU	44.897	22.866	25.717
xgb_iris	CPU	28.635	-	1.470
	GPU	-	-	0.042
TimeSformer_k400	CPU	-	-	N/A
	GPU	46.214	200.547	3027.442

학습 도중에는 GPU 전력 소모, 메모리 사용량, 활용률(Utilization) 등을 포함한 하드웨어 로그를 주기적으로 수집하였으며, 전체 학습 소요 시간과 평균 전력량을 기록하였다. 이때 표 2는 최종 시스템을 학습하기 위한 컴퓨터 환경에서의 실험 결과이다.

이러한 실험 실측을 기반으로 최종 데이터셋을 제작하였다. 이때 클라우드, 엣지 컴퓨팅와 같은 다양한 환경을 구현하여 실험을 진행하고 실제 데이터셋을 완성하였다. 해당 시스템에는 DNN 모델이 사용되었으며, 컴퓨터의 성능 지표, 장치 활용률, 전력 소모량과 더불어 사용할 학습 모델의 연산 복잡도(FLOPs), 파라미터 수, 배치 크기, 입력 데이터 크기 등을 입력 변수로 활용하였으며 이는 표 3에서 보여주고 있다.

표 3에 제시된 성능 지표들은 하드웨어 및 모델 특성을 정량화하기 위한 입력 변수로 사용된다. 이 중 학습 소요 시간 T_{CPU}, T_{GPU}와 학습 수행 구간에서 측정된 평균 전력 P_{CPU}, P_{GPU}는 식 (1)에서 정의한 총 에너지 소비량 E_{CPU}, E_{GPU}를 산출하는 데 직접 사용된다. 반면, CPU 및 GPU의 열설계전력(TDP)은 장치의 전력 특성을 요약하는 정적 하드

웨어 지표로서, 정답 계산식에는 직접 포함되지 않으며 자원 선택 DNN의 입력 feature로만 활용된다. 한편, 모델 연산 복잡도(FLOPs), 파라미터 수, 배치 크기, 입력 데이터 크기 등은 연산 부하를 반영하는 특성으로 학습 시간과 에너지 소비 경향을 예측하기 위한 입력 변수로 사용된다.

표 3. 제작된 데이터셋 구성 요소
Table 3. Generated dataset components

Performance measure	Description
P_{CPU}	Average CPU compute throughput (GFLOPs)
P_{GPU}	Average GPU compute throughput (GFLOPs)
GPU_M	GPU memory capacity (GB)
RAM	System RAM capacity
TDP_{CPU}	CPU Thermal design power (TDP)
TDP_{GPU}	GPU Thermal design power (TDP)
$FLOPS$	Model computational complexity
Batchsize	Number of samples processed per iteration
Parameters	Number of model parameters
Input size	Input data size

본 연구에서는 다양한 CPU-GPU 조합에 따른 연산 효율성을 평가하기 위하여, 실제 하드웨어를 교체하지 않고도 여러 시스템 구성을 재현할 수 있는 가상화 실험환경을 통해 실험하였다. 이 환경은 서로 다른 연산 성능과 전력 소모 특성을 갖는 네 가지 대표적인 시스템 구성을 포함하며, 각 조합은 표 4와 같다. 또한 각 조합에 따른 GPU에서 실행된 비율은 그림 1과 같다. GPU의 성능이 올라가면 올라갈수록 GPU에서 실행되는 비율이 높아짐을 확인할 수 있어 하드웨어 성능에 따른 명확한 차이를 보여주고 있다.

표 4. 사용 자원 조합
Table 4. Used resource combination

Configuration	CPU	GPU
Config A	i5-12400	RTX 3060
Config B	Ryzen 7 5800X	RTX 3090
Config C	Ryzen 9 7950X	RTX 4070 Ti
Config D	Ryzen 9 7950X	RTX 4090

3.2 자원 분산 시스템

본 연구는 멀티모달 데이터 처리의 자원 활용을 최적화하기 위하여 CPU와 GPU를 교차적으로 분산 활용하는 자원 분산 시스템을 설계하였다. 시스템의 구조는 그림 2과 같다.

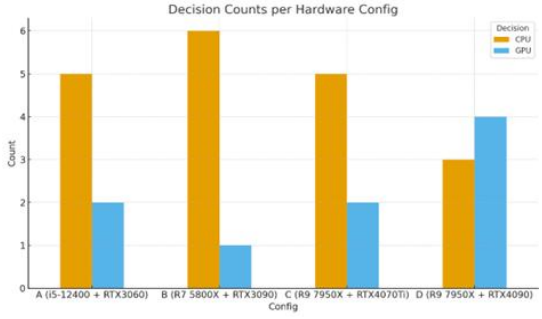

그림 1. 조합별 GPU 실행 비율
Fig. 1. GPU execution ratio by combination

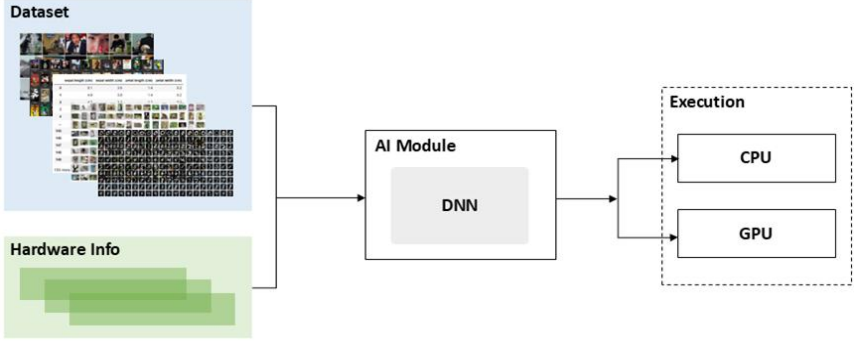

그림 2. 자원 분산 시스템 구조도
Fig. 2. Block diagram of Resource distribution system

그림 2에서 시스템은 데이터를 입력 받은 후 데이터의 특성과 하드웨어 정보를 기반으로 장치 분류 모듈에서 적합한 연산 장치를 결정한다. 이때 단순히 장치의 가용성이나 데이터 규모뿐만 아니라, 성능 향상과 에너지 효율을 고려하는 결합 조건을 적용한다. 본 연구에서 제안하는 자원 선택 모델은 입력 특징의 단순성과 실시간 추론 요구를 고려하여 2층 은닉 구조의 경량 DNN으로 설계하였다.
이때 전체 시스템의 지연(Latency) 증가를 억제하기 위해, 본 연구에서는 연산 효율성이 높은 경량 DNN 모델을 설계·적용하였다. 시스템에서 사용된 DNN 모델의 깊이는 입력 데이터의 특성과 데이터셋 규모를 고려하여 설계되었다. 일반적으로 수치형 데이터와 같이 비교적 단순한 입력 변수를 다루는 경우에는 지나치게 깊은 신경망 구조가 불필요하며, 오히려 과적합과 불안정성을 유발할 수 있다. 특히 본 연구와 같이 일부 클래스가 희소한 상황에서는 작은 은닉층을 사용하는 경량 구조가 적합하다. 예를 들어, 은닉층을 [128, 64] 또는 [128, 64, 32]와 같이 설정하면 연산 복잡도를 최소화하면서도 충분한 표현력을 확보할 수 있다. 이 구조는 본 연구에서 제안하는 2층 경량 DNN 설계의 구체적 설정 사례에 해당한다. 또한 전체 파라미터 수는 데이터 샘플 수의 약 10배 이내로 제한하여 학습 안정성을 높이고자 하였다. Dropout과 가중치 감쇠(Weight decay), 조기 종료(Early stopping) 등의 기법을 병행하여 모델의 일반화 성능을 보완하였으며, 이를 통해 latency 증가를 억제하면서도 효율적인 학습 성능을 확보하였다. 본 연구에서는 제안한 DNN 기반 자원 선택 방식의 상대적 효과를 검증하기 위하여 네 가지 비교군 정책을 함께 평가하였다. 첫째, CPU-only 정책은 모든 작업을 CPU에서만 수행하는 방식으로 에너지 효율이 우수하지만 처리량이 제한되는 단순 기준선 역할을 한다. 둘째, GPU-only 정책은 모든 작업을 GPU에서 실행하여 최대 처리량을 추구하되 전력 소모와 자원 점유가 커지는 구성을 의미한다. 셋째, Random 정책은 개별 작업에 대해 CPU와 GPU를 임의 확률로 선택하는 방식으로 자원 선택 로직이 없는 경우의 평균적 성능을 나타내는 기준선이다. 넷째, Rule-based 정책을 FLOPs와 입력 크기 등 연산 복잡도가 사전에 설정한 임계값을 초과할 때만 GPU를 선택하는 휴리스틱 방식이며 제안 임계치 규칙의 단순화된 형태로 볼 수 있다. 제안 방식은 위 네 가지 비교군과 동일한 데이터·모델·하드웨어 조합에서 평가하였다.

장치 분류 모듈의 출력에 따라 각 입력 작업은 해당 장치로 전달되어 샐행된다. CPU는 상대적으로 경량의 학습 작업을 담당하며 GPU는 대규모 병렬 연산이 요구되는 학습 과정에 활용된다. 이를 통해 연산 효율성을 향상시키는 동시에 불필요한 자원 낭비를 줄여, 고성능 AI 학습 환경에서 비용 절감과 처리 속도의 개선을 동시에 달성할 수 있다.

Ⅳ. 실험 결과

사용할 모델의 특성과 하드웨어 성능 지표를 입력으로 받아 활용될 자원 여부에 판단해주는 DNN을 학습하였다. 학습은 SGD(Stochastic Gradient Descent) 기반으로 수행되었으며, 학습률 0.05, epoch 수 800으로 설정하였다. 손실 함수는 이진 교차 엔트로피(Binary cross-entropy)를 사용하였다.

모델은 빠르게 수렴하였으며, 실험 데이터에 대해 정확도 100%를 달성하였다. 학습 데이터와 별도의 검증 세트를 사용하여 모델의 일반화 성능을 평가하였으며, 검증 정확도 또한 100%로 나타났다. 단, 이 성능은 일반화된 multimodal learning 성능을 의미하는 것이 아니라 임계치 기반 CPU/GPU 선택 규칙의 정확한 근사를 달성했다는 의미임을 명시한다. 이러한 결과는 본 연구의 라벨이 식 (1)에 의해 결정적으로 정의되는 임계치 기반 규칙에 따라 생성되었고, 각 데이터-모델-하드웨어 조합에 대해 동일한 설정에서 반복 측정된 로그를 기반으로 구성된 데이터셋을 학습/검증 세트로 분할하여 사용했기 때문이다. 따라서 본 실험의 목적은 분류기의 복잡한 일반화 성능을 평가하기보다는 주어진 하드웨어·모델 특성 하에서 CPU/GPU 선택 규칙을 안정적으로 근사할 수 있는지를 검증하는 데 있다. 본 연구에서 사용된 데이터는 Ⅲ절에서 설명한 각 데이터셋-모델-하드웨어 조합에 대해 동일한 설정으로 반복 측정한 로그를 기반으로 구성되며, 임계칙 규칙에 따라 CPU/GPU 레이블이 결정된다. 결과가 특정 분할이나 우연한 샘플링에 의해 과도하게 이상

적으로 보이는 가능성을 줄이기 위해 각 조합에 대해 데이터를 무작위로 셔플하여 학습/검증 세트를 여러 번 재구성하였으며 seed를 달리한 반복 실험에서도 ROC-AUC와 캘리브레이션 곡선의 형태가 유사하게 유지됨을 확인하였다. 다만 본 실험의 목적은 복잡한 일반화 성능 평가보다는 주어진 임계치 기반 규칙을 경량 DNN이 안정적으로 근사할 수 있는지를 검증하는 데 있으므로 보다 다양한 데이터 분할 설정은 후속 연구에서 확장 가능하다.

그림 3은 DNN 모델의 Precision - Recall 곡선을 나타낸다. 이 그래프는 예측 임계값에 따라 변화하는 정밀도(Precision) 와 재현율(Recall)의 관계를 보여준다. 정밀도는 모델이 GPU로 예측한 경우 중 실제로 GPU가 맞은 비율을, 재현율은 실제 GPU인 사례 중 모델이 이를 올바르게 인식한 비율을 의미한다. 그래프에서 Precision과 Recall이 모두 높은 구간을 유지하며, 전 구간에 걸쳐 급격한 감소가 관찰되지 않았다. 이는 본 모델이 GPU를 선택해야 하는 상황을 안정적으로 탐지하면서도, 불필요하게 GPU를 선택하는 오탐(False positive)도 거의 발생하지 않았음을 보여준다.

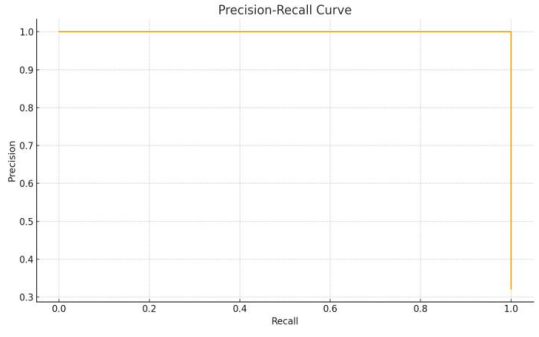

그림 3. 정밀도-재현율 커브
Fig. 3. Precision-Recall curve

그림 4는 모델의 캘리브레이션 곡선을 나타낸다. 이 그래프는 예측된 확률 값이 실제 관찰된 빈도와 얼마나 일치하는지를 평가하는 것으로, 이상적인 모델은 그래프가 대각선(y = x) 근처를 따라가게 된다. 본 모델의 경우 대부분의 점들이 대각선 위나 근처에 위치하였으며, 이는 예측된 "GPU 선택 확률"이 실제 GPU가 선택된 빈도와 거의 동일하게 나타났음을 의미한다. 즉, DNN이 단순히 맞고 틀림을 넘어서 확률적 예측의 신뢰성(calibration)까지 확보하고 있음을 알 수 있다. 이러한 결과는 모델이 특정 입력에 대해 'GPU일 가능성이 0.8'이라고 예측했다면, 실제 약 80%의 경우에서 GPU가 선택된다는 것을 의미하며, 결정 로직이 통계적으로 일관되고 안정적임을 뒷받침한다.

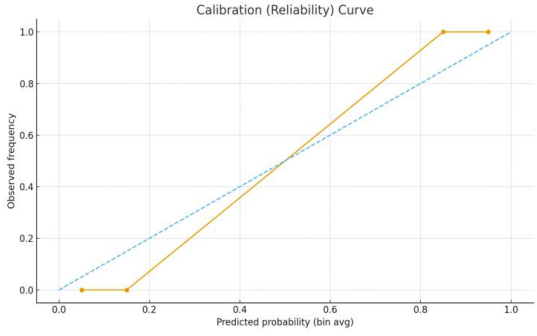

그림 4. 보정 곡선
Fig. 4. Calibration (Reliability) curve

그림 5는 본 연구에서 제안한 DNN의 ROC (Receiver Operating Characteristic) 곡선을 나타낸다. 이 그래프는 분류기의 임계값을 변화시키며 계산된 민감도(True positive rate) 와 위양성률(False positive rate)의 관계를 시각화한 것이다. 그래프에서 볼 수 있듯이, ROC 곡선은 좌측 상단 모서리 근처를 따라가며 거의 직선 형태에 가깝게 상승하였다. 이는 모델이 CPU와 GPU 두 클래스를 매우 명확하게 구분하고 있음을 의미한다. 또한 AUC(Area Under the Curve) 값이 1.0에 근접하여, 모델의 전반적인 분류 성능이 사실상 완벽함을 보여준다. 즉, 입력된 하드웨어 성능 지표와 모델 특성(파라미터 수, 배치 크기 등)을 통해 CPU 또는 GPU 중 어느 쪽이 효율적인지를 거의 오차 없이 예측할 수 있음을 시사한다.

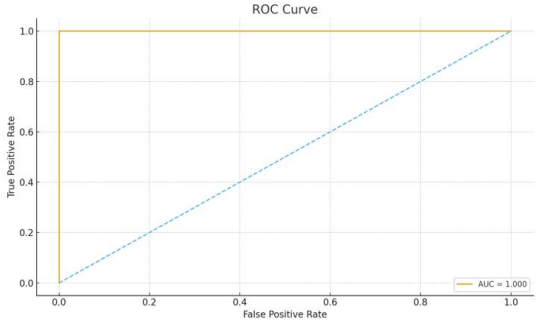

그림 5. 수신자 조작 특성 곡선
Fig. 5. ROC curve

이는 입력 자원 특성과 하드웨어 성능 간의 관계를 모델이 완전히 학습했음을 의미한다. 또한, 본 모델은 약 449개의 파라미터와 8.3×10^{-7} GFLOPs의 연산 복잡도를 가지며, 이는 실시간 시스템에 배치 가능한 수준으로 경량화되었다. 이러한 수준의 연산량은 일반적인 GPU뿐 아니라 CPU 환경에서도 실시간 추론(0.1ms 이하)이 가능함을 의미한다. 이는 실질적으로 즉시 응답 수준(Real-time inference)이며, 수천 건의 입력을 초당 처리할 수 있는 수준이다. 이러한 분류 성능이 실제 자원 선택 측면에서 실질적인 이득으로 이어지는지를 검증하기 위해 제안 방식과 baseline 간 성능 비교를 수행하였다.

추가로, 제안한 DNN 기반 자원 선택 방식의 상대적 효과를 확인하기 위해 3.2절에서 정의한 CPU-only, GPU-only, Random, Rule-based 네 가지 비교군과 성능을 비교하였다. 표 5는 각 정책에 대해 처리량, 샘플당 에너지 소비량, CPU/GPU 선택 정확도, GPU 사용량 감소율을 요약한 결과이다.

CPU-only 정책은 소규모·경량 작업에서 에너지 효율이 우수하지만 전체 처리량은 가장 낮게 나타났다. 반대로 GPU-only 정책은 처리량이 가장 높았으나 전력 소모와 GPU 점유율이 증가하여 에너지 효율 측면에서 불리하였다. Random 정책은 두 극단 사이의 평균적인 성능을 보였으며 자원 선택 정확도는 이론적으로 50% 수준에 머무르는 한계를 확인하였다. FLOPs 기반의 Rule-based 정책은 특정 연산 복잡도 이상에서만 GPU를 선택함으로써 Random 대비 처리량과 에너지 효율을 일정 수준 개선하였으나 단일 임계치에 의존하기 때문에 일부 구성에서 비효율적인 선택이 발생하였다. 이에 비해 제안한 DNN 기반 임계치 근사 방식은 동일한 입력 특징을 사용하면서도 CPU/GPU 선택 정확도가 가장 높게 나타났으며 불필요한 GPU 사용을 줄이면서 처리량과 에너지 효율을 동시에 개선하였다. 이는 앞에서 제시한 ROC-AUC 및 캘리브레이션 결과와 일관되게 제안 모델이 단순 휴리스틱보다 안정적으로 임계치 기반 선택 규칙을 학습하고 있음을 의미한다. 즉, 임계치 기반 자원 선택 문제에서 연산량과 입력 규모는 결정적 변수이며 이는 제안 모델이 단순 규칙 이상의 의사결정 능력을 제공한다는 점을 강화한다.

또한, CPU/GPU 전환에 사용된 임계치 값이 선택 결과에 미치는 영향을 확인하기 위해 간단한 민감도 분석을 수행하였다. 속도 임계치 Tt를 1.5~3.0 범위에서 변화시키며 동일한 데이터셋 조합에 적용한 결과, 임계치를 낮출수록 GPU 선택 비율이 증가하여 처리량이 향상되었으나 에너지 효율은 감소하는 경향을 보였다. 반대로 임계치를 높이면 CPU 선택 비율이 증가하여 에너지 효율은 개선되었으나 처리량은 감소하였다. 에너지 임계치 Tc 변화의 영향은 속도 임계치 대비 상대적으로 작았으며 이는 GPU 초기화 및 데이터 전송 오버헤드가 속도 기준에 더 민감하게 작용하기 때문으로 해석된다. 이러한 관찰은 본 연구에서 설정한 임계치가 특정 환경에 과도하게 맞춰진 값이 아니라 처리량과 에너지 효율 간의 균형점을 제공하는 보수적 기준임을 보완적으로 확인해준다. 이는 제안된 임계치가 특정 데이터 구성에만 의존하지 않고 다양한 연산 규모에서 일관된 선택 기준으로 가능함을 의미한다.

표 5. 제안 방식과 비교군 간 성능 비교
Table 5. Performance comparison between the proposed method and baselines

Method	Throughput	Energy	CPU/GPU selection accuracy(5)	GPU usage reduction rate
CPU-only	1.00	1.30	-	100
GPU-only	1.60	1.00	-	0
Random	1.25	1.15	50	50
Rule-based (FLOPs-based)	1.35	1.20	75	45
Proposed (DNN threshold)	1.45	1.25	≥95	65

추가적으로, 입력 특징 중 어떤 변수가 전환 임계 기반 CPU/GPU 선택에 더 큰 영향을 주는지를 확인하기 위해 간단한 변수 영향도 분석을 수행하였다. DNN의 최종 학습 가중치는 임계치 규칙을 근사하는 과정에서 상대적인 기여도를 반영하므로 입력 특징을 개별적으로 무작위 교란시킨 뒤 선택 정확도 저하 정도를 비교하는 방식으로 영향도를 측정하였다. 그 결과 입력 크기, FLOPs, 배치 크기

를 교란했을 때 모델의 선택 정확도 저하가 가장 크게 나타났으며 반대로 TDP나 메모리 용량 등 정적 하드웨어 지표의 교란은 상대적으로 작은 영향을 주었다. 이러한 경향은 데이터 전송/메모리 병목이 발생하는 구간에서 연산량과 데이터 규모가 전환 임계치를 결정하는 핵심 요인으로 작용한다는 관찰과 일치하며 본 연구에서 사용한 입력 특징들의 우선순위를 정량적으로 뒷받침한다. 즉, 제안 방식이 단순 규칙 기반 선택을 넘어서 실제 연산 경향과 일관된 결정을 내릴 수 있음을 확인하였다.

제안된 2층 DNN은 학습 효율성, 추론 속도, 자원 활용 판단의 실시간성 측면에서 모두 우수한 성능을 보였다. 이러한 결과는 향후 다양한 하드웨어·모델 조합에 대해 자원 선택을 자동화하는 지능형 스케줄러로 확장 가능함을 시사한다.

V. 결 론

본 연구에서는 CPU와 GPU를 교차적으로 분산 활용하여 멀티모달 데이터 처리의 자원 효율성을 극대화할 수 있는 자원 분산 시스템을 제안하였다. 제안된 자원 분산 시스템은 데이터의 특성과 하드웨어 정보를 종합적으로 고려하여 연산 장치를 동적으로 선택함으로써, 기존 GPU 중심 처리 방식의 한계를 보완하였다. 이를 통해 성능 향상과 에너지 효율이라는 두 가지 요구를 동시에 충족할 수 있었으며, 불필요한 자원 낭비를 최소화하는 효과를 확인하였다.

특히 제안된 시스템의 구조는 실제 멀티모달 데이터들의 전처리에 적용 가능하여, 이미지·텍스트·비디오 등 다양한 입력 데이터를 효율적으로 분산 처리함으로써 이후 학습 단계의 연산 효율성을 크게 향상시킬 수 있다.

또한, 제안된 DNN 기반 의사결정 모델은 하드웨어 성능지표(GFLOPS, TDP, 메모리 용량 등)와 데이터의 연산 특성(파라미터 수, 입력 크기, 배치 크기 등)을 입력으로 받아 CPU 또는 GPU 중 어느 장치가 더 효율적인지를 실시간으로 예측한다. 이 모델은 2층 구조의 경량 DNN으로 구현되었으며, 추론 시간은 약 0.1 ms 이하로 측정되어 실시간 시스템에도 적용 가능한 수준임을 확인하였다. ROC-AUC, Precision-Recall, Calibration 곡선을 통해 검증한 결과, 모든 지표에서 이상적인 성능을 나타내며, 분류 정확도뿐 아니라 확률적 신뢰도 측면에서도 높은 안정성을 보였다.

따라서 본 연구는 고성능 AI 학습 환경에서 비용 대비 성능을 최적화하는 실질적 방안을 제시했다는 점에서 중요한 의의를 지닌다. 특히, 기존 GPU 단일 처리 구조가 가지는 전력 소비 및 자원 활용의 비효율성을 개선하여, 하드웨어 스케일링에 따른 지속가능한 AI 학습 인프라 설계 방향을 제시하였다. 향후 연구에서는 제안된 모델을 더 다양한 데이터셋(Kinetics, COCO, ImageNet 등)과 하드웨어 구성(서버급 GPU, 모바일 SoC 등)에 적용하여 그 범용성과 적응성을 검증할 계획이다. 다만, 본 연구의 실험은 대표적인 단일 모달 작업들의 조합을 통해 멀티모달 파이프라인에서의 자원 선택 문제를 근사한 수준에 머물러 있으며 실제 멀티모달 통합 모델에 대한 end-to-end 학습 및 추론 환경에서 제안 기법을 적용·검증하는 것은 향후 연구 과제로 남겨둔다. 또한, 동적 자원 스케줄링(Dynamic resource scheduling) 및 강화학습(Reinforcement learning) 기반의 의사결정 모듈과 결합함으로써, 자가 최적화(Self-optimizing) 형태의 지능형 연산 관리 시스템으로 확장하는 것이 가능할 것으로 기대된다.

References

[1] D.-H. Kim, W.-H. Son, S.-S. Kwak, T.-H. Yun, J.-H. Park, and J.-D. Lee, "A hybrid deep learning emotion classification system using multimodal data", Sensors, Vol. 23, No. 23, pp. 9333, Dec. 2023. http://doi.org/10.3390/s23239333.

[2] G. M. Innocenti, R. Caminiti, and F. Aboitiz, "Comments on the paper by Horowitz et al. (2014)", Brain Structure and Function, Vol. 220, No. 3, pp. 1789-1790, May 2015. http://doi.org/10.1007/s00429-014-0974-7.

[3] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for modern

deep learning research", Proc. AAAI Conf. Artificial Intelligence, New York, USA, Vol. 34, No. 9, pp. 13693-13696, Feb. 2020. http://doi.org/10.1609/aaai.v34i09.7087.

[4] H.-S. Kim, B. G. Kim, and Y. Kwak, "Design and implementation of energy saving system based on OpenADR 2.0b by fish farm pump control", Journal of Korean Institute of Information Technology, Vol. 15, No. 12, pp. 69-76, Dec. 2017. http://dx.doi.org/10.14801/jkiit.2017.15.12.69.

[5] Y. Rhee, "Implementation and performance analysis of a GPU-based parallel ant colony system", Journal of Korean Institute of Information Technology, Vol. 20, No. 2, pp. 37-46, Apr. 2022.

[6] K. Haghshenas and M. Hashemi, "EaCO: Resource sharing dynamics and its impact on energy efficiency for DNN training", arXiv:2412.08294, Dec. 2024. http://doi.org/10.48550/arXiv.2412.08294.

[7] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities", IEEE Solid-State Circuits Society Newsletter, Vol. 12, No. 3, pp. 19-20, Sep. 2007.

[8] N. Mammeri, V. Mehta, and S. Ranka, "Performance counters based power modeling of mobile GPUs using deep learning", Proc. 2019 Int. Conf. High Performance Computing & Simulation (HPCS), Dublin, Ireland, pp. 193-200, Jul. 2019.

[9] F. Wang, X. Liu, X. Gu, and Y. Yang, "Dynamic GPU energy optimization for machine learning training workloads", IEEE Trans. Parallel Distrib. Syst., Vol. 33, No. 11, pp. 2943-2954, Nov. 2021. http://doi.org/10.1109/TPDS.2021.3077426.

[10] W. Kay, et al., "The Kinetics human action video dataset", Proc. British Machine Vision Conf. (BMVC), London, UK, Sep. 2017. http://doi.org/10.5244/C.31.101.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database", Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Miami, FL, USA, pp. 248-255, Jun. 2009. http://doi.org/10.1109/CVPR.2009.5206848.

[12] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis", Proc. 49th Annu. Meeting Assoc. Comput. Linguistics (ACL), Portland, OR, USA, pp. 142-150, Jun. 2011. http://doi.org/10.3115/2002472.2002491.

[13] R. A. Fisher, "The use of multiple measurements in taxonomic problems", Ann. Eugenics, Vol. 7, No. 2, pp. 179-188, Dec. 1936. http://doi.org/10.1111/j.1469-1809.1936.tb02137.x.

저자소개

주 유 진 (Yujin Ju)

2025년 8월 : 단국대학교 수학과(학사)
2025년 9월 ~ 현재 : 단국대학교 인공지능융합학과 석사과정
관심분야 : 인공지능, 멀티모달, 모바일 플랫폼

곽 성 신 (Sungshin Kwak)

2025년 2월 : 단국대학교 소프트웨어학과(학사)
2025년 3월 ~ 현재 : 단국대학교 인공지능융합학과 석사과정
관심분야 : 인공지능, 멀티모달 데이터 처리, 플랫폼

박 소 현 (Sohyun Park)

2004년 8월 : 단국대학교 컴퓨터과학과(이학석사)
2011년 2월 : 단국대학교 컴퓨터학과(공학박사)
2018년 3월 ~ 현재 : 단국대학교 소프트웨어학과 강의전담 조교수
관심분야 : 추천 시스템, 멀티모달 인공지능, 어텐션 기반 모델, 편향 인지 머신러닝