Journal of KIIT. Vol. 23, No. 9, pp. 97-104, Sep. 30, 2025. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2025.23.9.97

온디바이스 기반 GeneFace++의 성능향상을 위한 방안 연구

이재만*, 김선종**

A Study on Performance Improvement of On-Device GeneFace++

Jae-Man Lee*, Seon-Jong Kim**

요 약

최근 온디바이스 시스템의 발전은 점점 더 로컬에서 추론 처리를 수행하는데 중점을 두고 있다. 서버-클라이언트 방식은 높은 서버의 부하 및 민감한 개인 정보의 우려로 인하여 이용성 저하로 시스템의 유용성을 감소시킬 수 있다. 본 연구에서는 얼굴 합성 모델인 GeneFace++의 핵심 처리 단계 중 하나인 오디오를 얼굴 모션으로 변환할 때, 음성의 특징추출 과정에서 많은 시간이 소요된다는 문제를 확인했다. 이를 해결하기 위해 FastHuBERT와 추론 최적화 방법(ONNX)을 적용하여 처리시간을 크게 단축하는 방법을 제안한다. 실험 결과, 온디바이스 환경에서 얼굴 합성 모델의 추론 시간을 최대 20.31배 단축되었으며, 이를 통해 온디바이스에서도 얼굴 합성 모델과 같은 복잡한 추론을 효과적으로 수행할 수 있음을 확인하였다.

Abstract

Recent advancements in on-device systems increasingly focus on performing inference processing locally. Server-client architectures can suffer from reduced usability due to high server load and concerns regarding sensitive personal information. In this study, we identified that the audio feature extraction process, a crucial step in converting audio to facial motion within the face synthesis model GeneFace++, consumes a significant amount of processing time. To address this issue, we propose a method that significantly reduces the processing time by applying FastHuBERT and an inference optimization technique using ONNX. Experimental results demonstrate that the inference time of the face synthesis model in an on-device environment was reduced by up to 20.31 times. This confirms that complex inference tasks such as face synthesis models can be effectively executed on-device.

Keywords
geneface++, ondevice, processing time, fasthubert, onnx

* 부산대학교 IT응용공학과 박사과정
- ORCID: https://orcid.org/0000-0002-9685-3870
** 부산대학교 IT응용공학과 교수(교신저자)
- ORCID: https://orcid.org/0000-0003-2070-290X

· Received: Jun. 25, 2025, Revised: Jul. 28, 2025, Accepted: Jul. 31, 2025
· Corresponding Author: Seon-Jong Kim
Dept. of IT Engineering, Pusan National University, Korea
Tel.: +82-55-350-5413, Email: ksj329@pusan.ac.kr

Ⅰ. 서 론

인공지능 기술은 이미지 생성 및 변형 분야에서 급속한 혁신을 이끌고 있으며, 특히 얼굴 합성(Face synthesis)은 최근 가장 주목받는 연구 분야 중 하나다. 이러한 기술 발전의 핵심에는 생성적 적대 신경망(GAN, Generative Adversarial Networks)을 2014년 GAN을 도입하여, 생성자와 판별자의 경쟁 구조를 통해 사실적인 이미지 생성을 가능하게 하였다[1]. 이후 얼굴 합성 분야에서는 고해상도의 사실적인 이미지와 특히 사람얼굴 생성에 뛰어난 성능을 보이는 StyleGAN(Style-based Generative Adversarial Network), SPADE(Spatially-Adaptive Denormalization), Few-shot GAN 등 다양한 확장 연구가 진행되며 얼굴 이미지 생성의 품질과 다양성이 크게 향상되었다[2]-[4].

이러한 GAN 기반 얼굴 합성 기술은 단순한 이미지 생성에서 벗어나, 기존 얼굴 이미지를 기반으로 새로운 얼굴을 창조하거나 섬세하게 변형하는 것을 가능케 하였다. 이를 통해 영화 및 게임 분야의 영상 편집, 개인화된 디지털 아바타, 디지털 휴먼 제작뿐만 아니라, 신원 보호, 법의학, 의료 영상 분석 등 다양한 응용 분야에서 활용되고 있다[5].

최근 GeneFace(Generalized and High-Fidelity Audio-Driven 3D Talking Face Synthesis)++와 같은 모델은 음성의 입력으로고해상도 얼굴 이미지를 생성할 수 있어, 얼굴합성 기술의 가능성을 한층 확장시켰다. GeneFace++는 정밀한 입 모양 동기화, 미세한 표정 변화, 사실적인 피부 질감 구현이 가능하며, 고정밀의 음성 기반 3D 얼굴 애니메이션 생성을 지원한다[6].

하지만 이러한 고성능 모델은 방대한 연산량과 메모리 사용량을 요구한다. 대부분 고성능 GPU 서버 또는 클라우드 환경에 의존하고 있어, 실시간 상호작용 지연, 네트워크 종속성, 데이터 보안 문제 등 실용적 한계를 야기한다[7]. 특히 모바일 기기나 웨어러블 기기와 같은 온디바이스(On-Device) 환경에서는 제한된 연산 능력, 메모리, 전력 조건으로 인해 실시간 실행이 어렵다[8].

이러한 한계를 극복하기 위해 딥러닝 모델 경량화에 대한 다양한 연구가 수행되어 왔다. 대표적으로 모델 가지치기(Pruning)와 양자화(Quantization), 연산 병렬화, 메모리 접근 최적화, 하드웨어 특화 구조 설계 등이 있다[9]-[12]. 또한 MobileNet(Mobile Neural Network), MobileNetV2 등 경량 네트워크 구조들은 제한된 연산 환경에서도 효율적인 실행이 가능함을 보여주었다[13][14].

하지만 기존 연구는 주로 단일 기법(예: 모델 크기 축소)에 초점을 맞춰, GeneFace++와 같은 복잡하고 고정밀 모델의 온디바이스 실시간 구현에는 한계가 있다. 본 연구에서는 GeneFace++의 음성 기반 모션 생성 과정에 사용되는 음성에서 HuBERT (Hidden-Unit Bidirectional Encoder Representations from Transformers)특징 추출에 대해 Fast-HuBERT를 적용함으로써 처리시간을 대폭 단축하였고, Jetson Nano, Apple M4, NVIDIA RTX GeForce 등 다양한 디바이스에서의 성능을 실증적으로 평가하였다[15].

본 논문에서는 복잡한 다단계 구조를 갖는 GeneFace++ 모델의 효율성을 높이기 위해, 핵심 처리 단계 중 하나인 오디오 특징 추출 과정을 FastHuBERT와 ONNX(Open Neural Network Exchange)를 적용하여 최적화하는 방법을 제안하였다.

Ⅱ. 얼굴 합성 인공지능

2.1 GeneFace++

그림 1은 음성 입력(HuBERT Features)과 무작위 잠재 변수(z)를 활용하여 사람의 얼굴 움직임(랜드마크)을 생성하는 GeneFace++의 핵심 모듈이다.

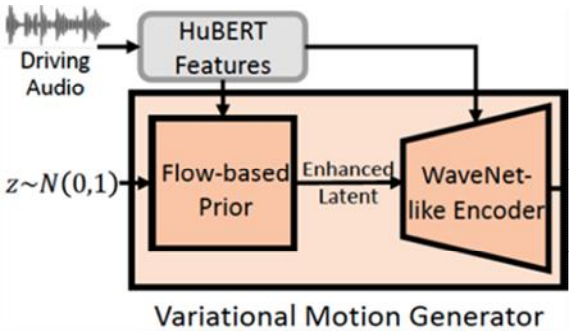

그림 1. GeneFace++의 음성특징 추출 구성
Fig. 1. GeneFace+++ audio feature extraction components

잠재 변수를 사용하여 다양한 움직임 생성을 가능하게 하고, 오디오 특징을 통해 생성된 움직임이 음성에 동기화하고 WaveNet-like Encoder는 이러한 입력들을 효과적으로 통합하여 고품질의 랜드마크 시퀀스를 생성하게 된다.

2.2 제안하는 방법

우리는 GeneFace++ 얼굴합성 모델에서 음성특징을 추출하는 부분을 개선하고자 음성특징 부분의 모델구성을 그림 2에 나타내었고 HuBERT부분을 FastHuBERT와 ONNX의 구성으로 교체하는 것을 제안한다. 기존에 HuBERT에 비해 FastHuBERT의 주요 차이점은 HuBERT의 표현 학습능력을 유지하면서 모델의 깊이, 넓이, 어텐션과 다양한 측면에서 구조적 조정을 통해 파라미터 수를 줄이고 불필요한 연산을 최소화하여 효율성을 높인 모델이다. 또한, ONNX는 다양한 딥러닝 프레임워크간에 모델을 상호 운용 가능하도록 설계된 개방형 표준 포맷으로써 다양한 프레임워크에 종속되지 않으며 여러 하드웨어 백엔드(CPU, GPU, NPU)에서 효율적으로 실행할 수 있도록 지원하며 배포의 용이성으로 온디바이스에서 구동하기에 좋고 변환과정에서 최적화를 통해 성능을 더욱 향상시킬 수 있다. 이 방법들의 조합을 통해 전체 얼굴합성 처리과정에서 가장 많은 시간을 소모하는 부분을 개선함으로써 전체 합성모델의 처리시간을 단축하여 온디바이스에서의 처리를 좀 더 원활하게 동작할 수 있다.

III. 실험 결과 및 분석

3.1 실험 환경 및 병목 분석

실험에 사용한 디바이스의 목록을 표 1에 나타내었다. Jetson Nano와 같은 저전력과 저사양을 가진 온디바이스와 일반적으로 사용되는 CUDA 프로세서가 없는 Apple M4, 제품군 중에서 메인스트림급의 RTX 4060, 그리고 서버급에 사용되는 RTX A5000 등으로 다양한 포지션을 가지는 디바이스에서 성능평가를 진행하였다.

테스트 음성 데이터는 50개로써 샘플링 레이트는 20050Hz에 최소 2.37초, 최대 10.73초에 평균 길이는 4.64초로 이루어져 있다. 표 2은 우선 Jetson Nano에서 GeneFace++모델에서 어느 부분이 가장 처리시간이 소요되는지 확인하기 위해 실험한 결과이다.

표 1. 실험에 사용된 디바이스 목록
Table 1. Devices used in the experiment

Device	TOPS (INT8)	Wattage	Release year
Jetson Nano	0.5 TOPS	~10W	2019
RTX A5000	40 TOPS	~230W	2021
RTX 4060	24 TOPS	~115W	2023
Apple M4	38 TOPS	~65W	2024

표 2. Jetson Nano에서 음성파일 50개의 평균 처리시간 결과
Table 2. Average processing time for 50 audio files on jetson nano

Process	Average time(sec)	Average rate(%)
Audio conversion	2.01	0.30%
HuBERT extraction	465.06	68.99%
F0 feature extraction	18.41	2.73%
Batch preparation	170.44	25.29%
Audio to motion	18.14	2.69%
Total processing time	787.7	100%

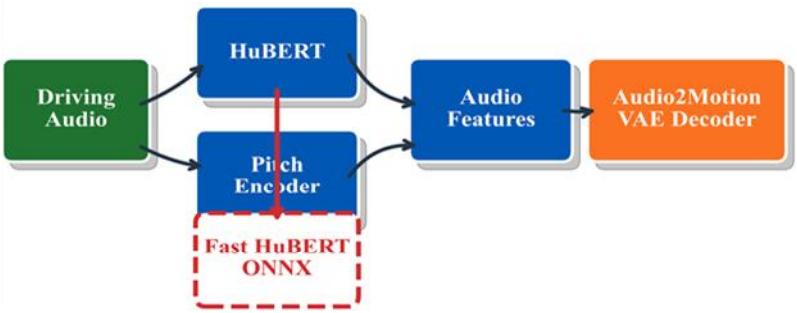

그림 2. GeneFace++을 위해 제안하는 오디오 특징 추출 방법
Fig. 2. Proposed audio feature extraction method for GeneFace++

실험 결과를 분석해보면 50개의 음성샘플의 평균을 취하였고 세부적인 병목 분석 결과, 전체 처리시간의 약 68.99%가 HuBERT 특징 추출 단계에서 소요되어 이 단계가 가장 큰 병목 지점임이 확인되었다. 그 다음으로는 배치 준비 단계가 25.29%를 차지하여, 데이터셋 로딩 및 전처리 과정에서 상당한 시간 소모가 발생하고 있음을 보여주었다. 오디오 신호를 모션 데이터로 변환하는 단계는 평균적으로 2.69%의 비중을 차지하였으나, 일부 길이가 긴 특정 음성에서는 최대 22.38%까지 상승하는 등 처리 비율의 편차가 큰 것으로 나타났고 이것은 Jetson Nano이 기본적으로 SD Card를 저장 장치로 쓰고 CPU의 성능이 낮음에서 나타난 현상으로 보여진다.

3.2 최적화 방법 적용 결과

이러한 분석 결과를 바탕으로, 처리 성능 개선을 위해 다음과 최적화 방안을 제안하였다. 먼저, 음성 특징 추출 단계를 FastHuBERT로 교체하고 ONNX 형식으로 변환하는 방법으로써 특히, Jetson Nano와 같이 연산 자원이 제한된 환경에서는 HuBERT 특징 추출 단계를 최우선적으로 최적화하는 것이 전체 시스템 성능을 향상시키는 데 가장 효과적인 전략으로 판단하였다.

표 3는 다양한 디바이스에서 HuBERT와 FastHuBERT 모델을 ONNX로 변환하여 적용했을 때의 평균 처리시간(50개 음성 샘플)을 보여준다. 분석 결과, FastHuBERT 모델에 ONNX 변환을 적용했을 때 가장 뚜렷한 성능 향상을 확인할 수 있었다.

표 3. 다양한 컴퓨팅 아키텍쳐에서 원본 및 최적화된 HuBERT 모델의 성능 평가(초)
Table 3. Performmace evaluation of original and optimized HuBERT models on various computing architectures(sec)

Device	HuBERT	HuBERT (ONNX)	Fast HuBERT	Fast HuBERT (ONNX)
Jetson Nano	465.07	371.42 (1.25x)	220.84 (2.11x)	32.30 (14.40x)
RTX A5000	3.39	1.99 (1.71x)	1.51 (2.25x)	0.18 (18.62x)
RTX 4060	3.91	1.72 (2.27x)	1.17 (3.33x)	0.19 (20.31x)
Apple M4	1.57	1.76 (0.89x)	0.66 (2.36x)	0.28 (5.45x)

특히 저사양 디바이스인 Jetson Nano에서는 FastHuBERT (ONNX)가 원본 HuBERT 대비 14.40배의 속도 향상을 보였다. 이는 제한적인 컴퓨팅 자원을 가진 환경에서 ONNX 변환을 통한 효율성 증대의 가능성을 나타낸다. 더욱 주목할 만한 점은 고성능 칩인 RTX A4060에서의 결과인데, FastHuBERT (ONNX)는 원본 HuBERT 대비 최대 20.31배 속도 향상을 나타냈다. 이는 고성능 환경에서도 최적화된 추론 엔진을 통해 모델의 잠재력을 최대한으로 끌어올릴 수 있음을 보여준다.

또한, 본 연구에서는 GeneFace++ 코드의 일부를 수정하여 Apple M4 칩의 GPU를 활용한 추론을 시도하였으며, 그 결과 원본 HuBERT 대비 5.45배의 성능 향상을 확인하였다. 이는 특정 하드웨어에 맞춘 최적화 역시 상당한 성능 향상을 가져올 수 있음을 보여준다. 전반적으로 ONNX 변환 모델은 4가지 장치에서 원본 모델 대비 평균 14.70배의 빠른 추론 속도의 효과를 보였다.

실험을 통해 HuBERT와 FastHuBERT는 self-supervised 음성 모델이지만 효율성 측면에서 뚜렷한 차이를 보임을 재확인하였다. HuBERT는 원시 음성 파형을 직접 처리하며 깊은 CNN과 복잡한 Transformer 구조(약 95M 파라미터)를 통해 높은 성능을 달성하지만, 상당한 계산 자원을 요구하고 처리 속도가 느리다. 반면, FastHuBERT는 Fbank 특징을 입력으로 사용하고 더 간단한 CNN과 작은 Transformer 구조(약 90M 파라미터)를 채택하여, 원시 파형 직접 처리의 부담 없이 빠른 추론 속도와 낮은 메모리 사용량을 달성했다. 이러한 특징은 향후 점진적으로 중요해질 온디바이스 환경에서의 실시간 음성 처리 애플리케이션에 FastHuBERT가 더욱 적합할 수 있음을 나타낸다.

3.3 종합 성능 분석

그림 3, 4은 얼굴합성의 결과에서 프레임별 얼굴의 랜드마크 특징을 비교해서 얼마나 차이가 나는지 실험을 하였다. 그 결과, 모델 별 그리고 ONNX를 적용했을 때 차이가 거의 없음을 확인할수 있었다. 평균 픽셀의 차이로 볼 때 실제 합성얼굴의 영상을 사람의 눈으로 볼때는 구분하기 어려울 정도의 적은 차이를 가지면서도 처리 성능은 향상되었다.

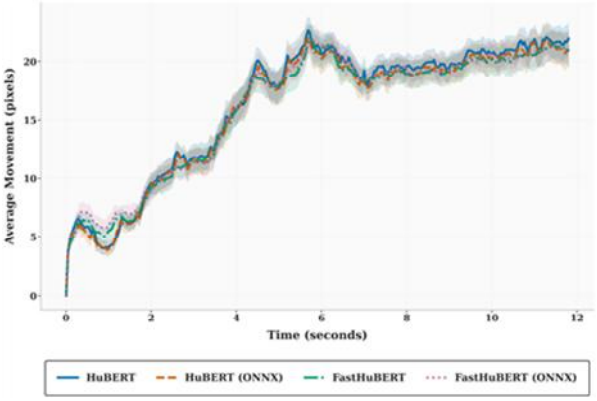

그림 3. 모델 간 얼굴표정 움직임 비교
Fig. 3. Facial movement comparison across models

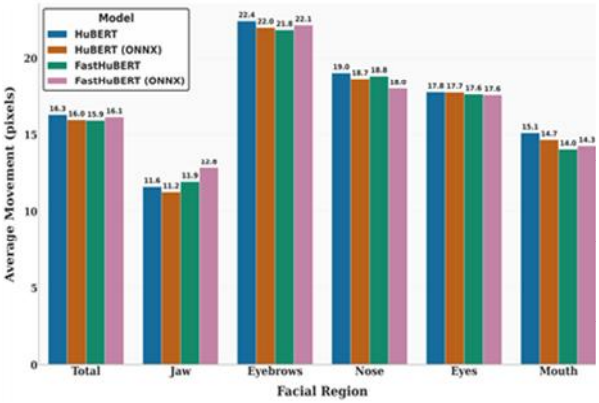

그림 4. 모델과 얼굴의 부위별 평균 움직임
Fig. 4. Average facial movement by model and region

또한, 각 모델과 ONNX 변환의 차이점을 좀 더 확인하기 위해서 추가적 실험에 사용한 음성 파일은 GeneFace++에 기본 예제로 들어있는 단일 음성파일로써 마크롱의 연설 중 약 12초의 음성파일로 비교하였다. 특히 중요한 점은 얼굴 움직임 생성 정확도에서 ONNX 변환 전후 통계적으로 유의미한 차이가 관찰되지 않았다는 것이다. 이는 ONNX 변환이 모델의 정확도 손실 없이 실시간 애플리케이션에 적용 가능한 효과적인 최적화 방법임을 입증한다.

본 연구에서는 ONNX 변환이 모델의 특징 표현에 미치는 영향을 분석하기 위해 각 모델의 특징 분포를 시각적으로 비교하였다. 그림 5는 원본 모델과 ONNX로 변환된 모델의 모든 특징값에 대한 밀도 분포를 나타낸다. 그래프의 세로축은 특징값의 밀도를 의미하며, 각 모델별로 다른 색상의 곡선이 특징값의 분포를 보여준다. 분포 곡선의 형태와 위치를 통해 모델이 추출하는 특징의 전반적인 경향성을 파악할 수 있다.

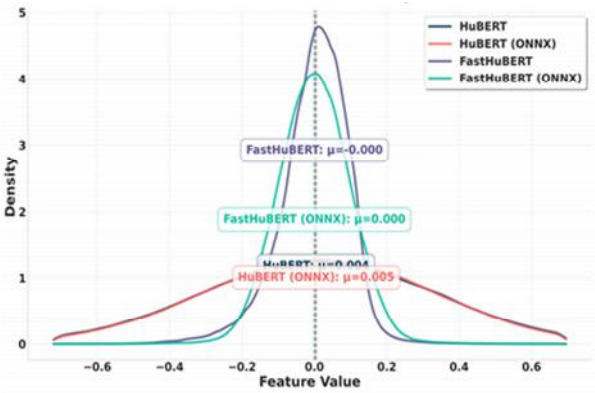

그림 5. 모든 특징 분포 비교
Fig. 5. All feature distributions comparison

각 분포 곡선 옆에 제시된 평균값 (μ)는 해당 특징 분포의 중심 경향을 나타낸다. HuBERT 모델의 평균값은 0.004, HuBERT(ONNX) 모델의 평균값은 0.005로, ONNX 변환이 HuBERT 모델의 특징 값 중심 위치에 미치는 영향은 미미한 것으로 나타났다. 이는 변환 과정에서 특징 값들이 전반적으로 유지되는 경향이다. 반면, FastHuBERT 모델과 FastHuBERT(ONNX) 모델은 모두 0.000의 평균값을 갖는다. 이는 FastHuBERT 모델 자체가 HuBERT 모델과는 다른 특징 값 분포의 중심을 가지며, ONNX 변환 역시 이러한 중심 경향을 보존함을 의미한다. 이는 ONNX 변환이 각 모델의 특징 분포의 전반적인 형태를 크게 변화시키지 않음을 시각적으로 뒷받침한다.

이러한 특징 분포 분석 결과는 ONNX 변환이 모델의 기본적인 특징 표현 능력을 유지하면서 효율성이 향상됨을 확인할 수 있다. 특히, 변환 전후의 특징 분포 유사성은 기능적 동등성을 확보하면서도 성능 최적화를 달성할 수 있음을 암시한다.

그림 6는 각 모델의 모델 크기와 GeneFace++에 포함된 단일 음성을 RTX 4060에서 처리시간을 막대 그래프로 나타낸 결과이다. 분석 결과, ONNX 변환은 HuBERT와 FastHuBERT 모델 모두에서 모델 크기를 유의미하게 감소시키는 것으로 확인되었다. HuBERT 모델의 경우 ONNX 변환 후 모델 크기가 약 32.1% 감소하였으며, FastHuBERT 모델 역시 유사하게 약 32.1%의 모델 크기 감소를 보였다. 이는 ONNX 포맷이 모델 구조를 효율적으로 변환함을 보여준다.

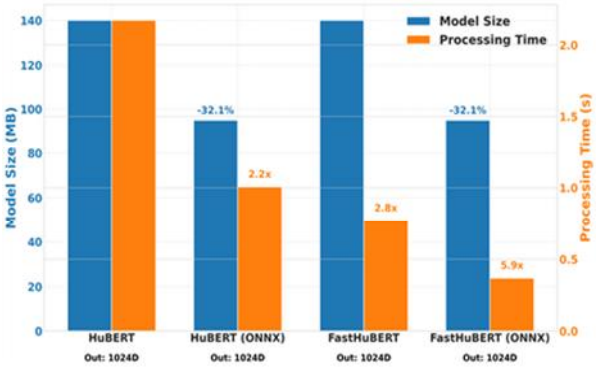

그림 6. 모델 크기 및 처리시간 비교
Fig. 6. Model size and processing time comparison

처리시간 측면에서 ONNX 변환은 더욱 뚜렷한 성능 향상을 나타냈다. HuBERT 모델의 처리시간은 ONNX 변환을 통해 약 2.2배 빨라졌으며, 특히 FastHuBERT 모델의 경우 ONNX 변환 후 처리시간이 5.9배 단축되는 개선을 보였다. 이는 ONNX Runtime이 모델을 실행하는 과정에서 다양한 최적화 기법을 적용하여 추론 속도를 크게 향상시키기 때문으로 해석될 수 있다.

그림 7은 가장 빠른 FastHuBERT를 ONNX 결과를 추론한 결과를 기반으로 추론 속도와 전력 대비 연산 효율을 비교한 것이다. Jetson Nano는 저전력(10W) 기반의 경량 장치로, 0.5 TOPS의 낮은 연산 성능을 가지고 있으므로 추론 시간이 32.30초로 가장 길었고, 전력 효율 또한 0.050 TOPS/W로 가장 낮게 나타났다.

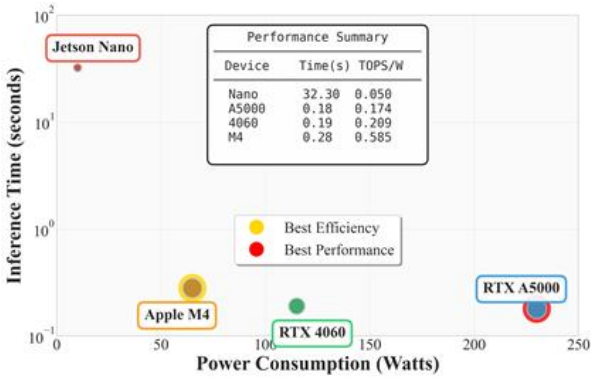

그림 7. 전력 대비 성능 효율 비교
Fig 7. Performance per watt across devices

반면, Apple M4는 65W의 전력 소비와 38 TOPS의 연산 성능으로, 0.28초의 빠른 추론 시간과 0.585 TOPS/W의 높은 전력 효율을 보여주었다. 이는 고성능 GPU 기반의 RTX 4060(0.19 TOPS/W)이나 RTX A5000(0.18 TOPS/W)에 비해서도 우수한 수치로, 최근 모바일 및 임베디드 장치에서 추론 특화 NPU가 효과적으로 활용되고 있음을 보여준다.

이러한 결과는 고성능 GPU가 높은 연산력을 가지더라도, 전력 대비 효율 측면에서는 온디바이스 추론을 위해 최적화된 SoC나 NPU 기반 장치들의 효율성을 보여준다. 특히, 최근의 추세는 학습보다 추론에 특화된 구조로 전환되고 있으며, 이에 따라 스마트폰, 태블릿, 임베디드 기기 등 다양한 디바이스에 추론 전용 NPU의 탑재가 활발히 이루어지고 있다. 이러한 방향은 향후 온디바이스 AI의 실용성과 에너지 효율을 더욱 높이는 긍정적인 흐름으로 작용할 것으로 기대된다.

결론적으로, 본 연구 결과는 ONNX 변환이 계산 자원이 제한적인 온디바이스 환경뿐만 아니라 고성능 환경에서도 딥러닝 모델의 성능을 효율적으로 향상시킬 수 있는 강력한 방법임을 보여준다. 특히 FastHuBERT 모델과 ONNX의 조합은 온디바이스 얼굴 합성 기술의 실시간 처리 가능성을 크게 높여 향후 상용화에 긍정적인 영향을 미칠 것으로 예상된다.

Ⅳ. 결 론

본 연구에서는 음성 기반 고해상도 3D 얼굴 애니메이션 생성 모델인 GeneFace++의 온디바이스 환경에서의 효율적인 구동을 목표로, 핵심 처리 단계인 오디오 특징 추출 과정에 FastHuBERT와 ONNX를 적용하여 성능 개선을 시도하였다. 기존 연구에서 GeneFace++ 모델의 오디오-얼굴 모션 변환 시 HuBERT 특징 추출 단계가 가장 많은 연산 시간을 소요하는 병목 지점임을 확인하고, 이를 해결하기 위해 경량화된 FastHuBERT 모델을 ONNX 포맷으로 변환하여 다양한 디바이스에서 성능 평가를 수행하였다.

실험 결과, 제안하는 FastHuBERT와 ONNX의 조합은 온디바이스 환경을 포함한 다양한 환경에서 GeneFace++ 모델의 추론 시간이 크게 단축됨을 확인하였다. 특히 연산 자원이 제한적인 Jetson Nano

환경에서 최대 14.40배, 중간급의 GPU인 RTX 4060 환경에서는 최대 20.31배의 추론 속도 향상을 달성하였다. 또한, Apple M4 칩의 GPU를 활용한 실험에서도 5.45배의 성능 향상을 확인하여, 특정 하드웨어 최적화의 가능성을 보여주었다.

모델 크기 및 처리시간 비교 분석 결과, ONNX 변환은 HuBERT와 FastHuBERT 모델 모두의 모델 크기를 약 32.1% 감소시켰으며, 처리시간은 평균적으로 1.8배 향상되었다. 특히 FastHuBERT 모델에 ONNX를 적용했을 때 가장 큰 폭의 처리시간 단축 효과를 얻을 수 있었다. 얼굴 움직임 생성 정확도 측면에서는 ONNX 변환 전후 유의미한 차이가 나타나지 않아, 제안하는 최적화 방법이 시각적 품질 저하 없이 효율성을 높이는 효과적인 전략임을 입증하였다.

본 연구를 통해 복잡한 얼굴 합성 모델의 특정 단계를 효율적으로 최적화함으로써 온디바이스 환경에서도 실시간에 준하는 속도로 추론을 수행할 수 있음을 확인하였다. 이는 향후 모바일 기기, 웨어러블 디바이스 등 다양한 온디바이스 환경에서 고품질의 실시간 음성 기반 얼굴 애니메이션 서비스를 제공하는 가능성을 열어줄 것으로 기대된다.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, and Y. Bengio, "Generative adversarial nets", Advances in Neural Information Processing Systems (NeurIPS), Montreal, Canada, pp. 2672-2680, Dec. 2014.

[2] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, pp. 4401-4410, Jun. 2019. https://doi.org/10.1109/CVPR.2019.00453.

[3] T. Park, M. Y. Liu, T. C. Wang, and J. Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, pp. 2336-2345, Jun. 2019. https://doi.org/10.1109/CVPR.2019.00244.

[4] Y. Cao and S. Gong, "Few-shot image generation by conditional relaxing diffusion inversion," arXiv preprint arXiv:2407.07249, Jul. 2024. https://arxiv.org/abs/2407.07249.

[5] Y. Zhang, Z. Wang, X. Wu, H. Zheng, and C. Zhang, "Visual content privacy protection: A survey," arXiv preprint arXiv:2303.16552, Mar. 2023. https://arxiv.org/abs/2303.16552.

[6] Z. Ye, Z. Jiang, Y. Ren, J. Liu, J. He, and Z. Zhao, "GeneFace: Generalized and High-Fidelity Audio-Driven 3D Talking Face Synthesis," arXiv preprint, arXiv:2301.13430, Jan. 2023. https://arxiv.org/abs/2301.13430.

[7] M. Satyanarayanan, "The emergence of edge computing", Computer, Vol. 50, No. 1, pp. 30-39, Jan. 2017. https://doi.org/10.1109/MC.2017.9.

[8] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications", arXiv preprint, arXiv:1704.04861, Apr. 2017. https://doi.org/10.48550/arXiv.1704.04861.

[9] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding", arXiv preprint, arXiv:1510.00149, Oct. 2015. https://doi.org/10.48550/arXiv.1510.00149.

[10] H. Li, V. Kadambi, J. M. Alvarez, and A. L. Yuille, "Pruning filters for efficient convnets", arXiv preprint, arXiv:1608.08710, Aug. 2016. https://doi.org/10.48550/arXiv.1608.08710.

[11] K. Chellapilla, S. Puri, and P. Y. Simard, "High performance convolutional neural networks for document processing", Proceedings of the International Workshop on Frontiers in Handwriting Recognition, La Baule, France, pp.

303-308, Oct. 2006.

[12] V. Sharma, P. Panda, V. Vasudevan, and K. Roy, "Low power optimization of deep convolutional neural networks using hardware aware pruning", Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, pp. 1-5, May 2018. https://doi.org/10.1109/ISCAS.2018.8351763.

[13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, pp. 4510-4520, Jun. 2018. https://doi.org/10.1109/CVPR.2018.00474.

[14] S. Eyerman and L. Eeckhout, "A survey of techniques and trends in memory system optimization", IEEE Transactions on Computers, Vol. 66, No. 12, pp. 2028-2045, Dec. 2017. https://doi.org/10.1109/TC.2017.2704600.

[15] P. C. Krause and O. Wasynczuk, "Fast-Hubert: an efficient training framework for self-supervised speech representation learning", 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Taipei, Taiwan, pp. 1-5, Dec. 2023. https://doi.org/10.1109/ASRU57964.2023.10389778.

저자소개

이 재 만 (Jae-Man Lee)

2011년 8월 : 부산대학교 바이오정보전자전공(공학사)
2014년 2월 : 부산대학교 IT응용공학과(공학석사)
2021년 3월 ~ 현재 : 부산대학교 IT응용공학과 박사과정
관심분야 : 신호 및 영상처리 머신/딥러닝

김 선 종 (Seon-Jong Kim)

1996년 8월 : 경북대학교 전자공학과(공학박사)
1995년 2월 ~ 1997년 2월 : 순천제일대학 전임강사
1997년 3월 ~ 현재 : 부산대학교 IT응용공학과 교수
관심분야 : 신호 및 영상처리, 머신/딥러닝, VR/AR, 스마트 카메라