



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0089927
(43) 공개일자 2023년06월21일

(51) 국제특허분류(Int. Cl.)
G06N 3/063 (2023.01) G06N 20/10 (2019.01)
G06N 3/08 (2023.01) G06V 40/16 (2022.01)
(52) CPC특허분류
G06N 3/063 (2013.01)
G06N 20/10 (2021.08)
(21) 출원번호 10-2021-0178711
(22) 출원일자 2021년12월14일
심사청구일자 2021년12월14일

(71) 출원인
인하대학교 산학협력단
인천광역시 미추홀구 인하로 100(용현동, 인하대학교)
상명대학교 천안산학협력단
충청남도 천안시 동남구 상명대길 31, 상명대학교내 (안서동)
(72) 발명자
강진구
서울특별시 서초구 양재대로2길 109, 104동 1202호(우면동, 서초호반써밋)
김재명
경기도 고양시 덕양구 푸른마을로 55, 605동 907호(고양동, 푸른마을6단지아파트)
김용우
서울특별시 송파구 백제고분로19길 3-22, 3층(잠실동)
(74) 대리인
양성보

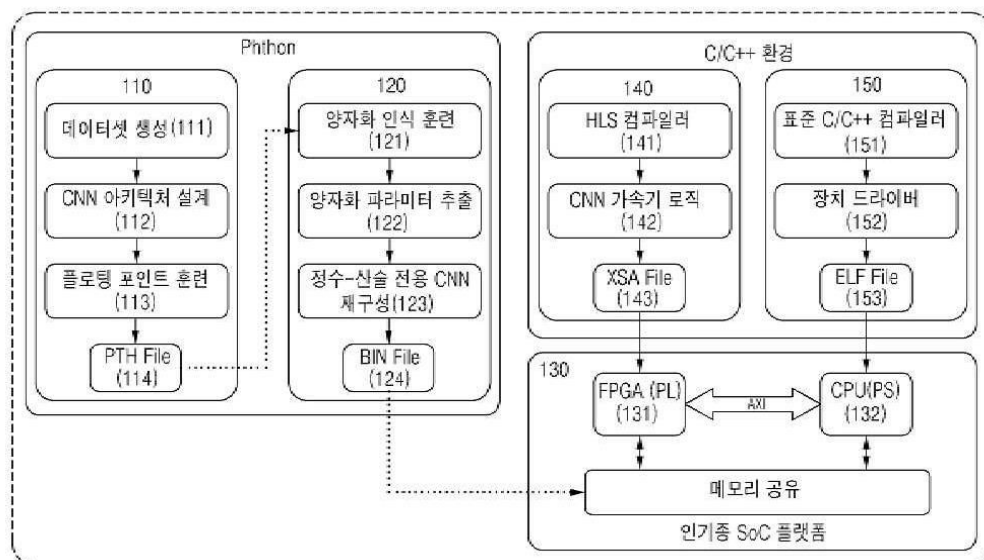
전체 청구항 수 : 총 8 항

(54) 발명의 명칭 실시간 얼굴 감정 인식을 위한 자원 효율적인 정수-산술 전용 FPGA 기반 CNN 가속기

(57) 요약

실시간 얼굴 감정 인식을 위한 자원 효율적인 정수-산술 전용 FPGA 기반 CNN 가속기 및 그 동작 방법이 제시된다. 본 발명에서 제안하는 CPU(Processing System; PS)와 FPGA(Programmable Logic, PL)를 포함하는 이기종 SoC 플랫폼에서 구현되는 FPGA 기반 CNN 가속기의 동작 방법은 CPU에서 CNN 얼굴 표정 인식 성능을 향상시키(뒷면에 계속)

대표도



기 위해 복수의 이미지 처리 알고리즘을 결합한 훈련 데이터셋을 생성하고, 생성된 훈련 데이터셋에 대하여 플로팅-포인트(floating-point) 훈련을 통해 파라미터를 변환하는 단계, CPU에서 플로팅-포인트 훈련을 통해 변환된 파라미터들에 대하여 양자화 인식 훈련을 통해 양자화된 파라미터를 추출하고, 정수-산술 전용(Integer-Arithmetic-Only) CNN 재구성을 수행하는 단계 및 이기종 SoC(System on Chip) 플랫폼의 FPGA(Programmable Logic; PL) 및 CPU(Processing System; PS) 영역에서 상기 정수-산술 전용 CNN을 이용하여 실시간 얼굴 감정 인식을 수행하는 단계를 포함한다.

(52) CPC특허분류

G06N 3/08 (2023.01)

G06V 40/174 (2022.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711139247
과제번호	2021-0-02052-001
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	대학 ICT연구센터 육성·지원사업
연구과제명	[Ezbaro][정부] 스마트 모빌리티를 위한 인공지능 시스템반도체 핵심 기술 개발 및
인력 양성 (1차년도)	
기 여 율	1/1
과제수행기관명	인하대학교 산학협력단
연구기간	2021.07.01 ~ 2021.12.31

명세서

청구범위

청구항 1

CPU(Processing System; PS)와 FPGA(Programmable Logic, PL)를 포함하는 이기종 SoC(System on Chip) 플랫폼에서 구현되는 FPGA 기반 CNN 가속기의 동작 방법에 있어서,

CPU에서 CNN 얼굴 표정 인식 성능을 향상시키기 위해 복수의 이미지 처리 알고리즘을 결합한 훈련 데이터셋을 생성하고, 생성된 훈련 데이터셋에 대하여 플로팅-포인트(floating-point) 훈련을 통해 파라미터를 변환하는 단계;

CPU에서 플로팅-포인트 훈련을 통해 변환된 파라미터들에 대하여 양자화 인식 훈련을 통해 양자화된 파라미터를 추출하고, 정수-산술 전용(Integer-Arithmetic-Only) CNN 재구성을 수행하는 단계; 및

이기종 SoC 플랫폼의 FPGA 및 CPU 영역에서 상기 정수-산술 전용 CNN을 이용하여 실시간 얼굴 감정 인식을 수행하는 단계

를 포함하는 FPGA 기반 CNN 가속기의 동작 방법.

청구항 2

제1항에 있어서,

상기 CPU에서 CNN 얼굴 표정 인식 성능을 향상시키기 위해 복수의 이미지 처리 알고리즘을 결합한 훈련 데이터셋을 생성하고, 생성된 훈련 데이터셋에 대하여 플로팅-포인트 훈련을 통해 파라미터를 변환하는 단계는,

제1 계산 모듈(FireA) 및 제2 계산 모듈(FireB)을 포함하는 기본 계산 블록을 통해 연산을 수행하고,

제1 계산 모듈(FireA) 및 제2 계산 모듈(FireB) 각각은,

특징맵의 채널 수를 감소시키고, 이어지는 확장 계층의 연산 수를 감소시키기 위한 스퀴즈 계층; 및

채널을 다시 확장하기 위한 확장 계층을 포함하고,

수신 필드를 유지하여 정확도 저하를 최소화하기 위해 제1 계산 모듈의 스퀴즈 계층과 제2 계산 모듈의 스퀴즈 계층은 서로 다른 커널 크기를 갖는

FPGA 기반 CNN 가속기의 동작 방법.

청구항 3

제2항에 있어서,

특징맵의 해상도를 절반으로 줄이고 최대 풀링 계층을 대체하기 위해 상기 기본 계산 블록의 스트라이드를 2로 설정하고,

CNN의 마지막에 추출된 특징을 분류하고 복수의 매개 변수를 포함하는 완전 연결 계층을 글로벌 평균 풀링으로 대체하고 커널 크기가 1인 컨볼루션 계층을 대체하여 상기 파라미터를 감소시키고,

CNN의 분류기는 컨볼루션 계층을 통해 클래스 수와 일치하도록 채널 수를 최소화한 후 특징맵의 해상도를 하나의 픽셀로 압축하며,

CNN의 수렴 안정성과 성능을 개선하기 위해 컨볼루션 계층과 활성화 함수 사이에 배치 정규화 계층을 삽입하는

FPGA 기반 CNN 가속기의 동작 방법.

청구항 4

제1항에 있어서,

상기 CPU에서 플로팅-포인트 훈련을 통해 변환된 파라미터들에 대하여 양자화 인식 훈련을 통해 양자화된 파라

미터를 추출하고, 정수-산술 전용 CNN 재구성을 수행하는 단계는,

훈련 가능한 파라미터를 2의 거듭제곱 인수에 매핑하고, 첫 번째 에포크(epoch)의 첫 번째 배치의 활성화 데이터 통계 값으로 활성화 기준화 인수를 초기화하여 사전 보정 프로세스를 필요로 하지 않고, 기준화 인수를 2의 거듭제곱 항에 매핑함으로써 스케일에서 발생하는 곱셈 및 나눗셈 연산과 역-양자화 프로세스는 모두 이동 연산으로 대체되는 로그레벨 임계값 정량화(Log Level Threshold Quantization; LLTQ) 방법을 이용하는

FPGA 기반 CNN 가속기의 동작 방법.

청구항 5

CPU(Processing System; PS)와 FPGA(Programmable Logic, PL)를 포함하는 이기종 SoC(System on Chip) 플랫폼에서 구현되는 FPGA 기반 CNN 가속기에 있어서,

상기 CPU는,

CNN 얼굴 표정 인식 성능을 향상시키기 위해 복수의 이미지 처리 알고리즘을 결합한 훈련 데이터셋을 생성하고, 생성된 훈련 데이터셋에 대하여 플로팅-포인트(floating-point) 훈련을 통해 파라미터를 변환하고,

플로팅-포인트 훈련을 통해 변환된 파라미터들에 대하여 양자화 인식 훈련을 통해 양자화된 파라미터를 추출하고, 정수-산술 전용(Integer-Arithmetic-Only) CNN 재구성을 수행하며,

이기종 SoC 플랫폼의 FPGA 및 CPU 영역에서 상기 정수-산술 전용 CNN을 이용하여 실시간 얼굴 감정 인식을 수행하는

FPGA 기반 CNN 가속기.

청구항 6

제5항에 있어서,

상기 CPU는,

CNN 얼굴 표정 인식 성능을 향상시키기 위해 복수의 이미지 처리 알고리즘을 결합한 훈련 데이터셋을 생성하고, 생성된 훈련 데이터셋에 대하여 플로팅-포인트 훈련을 통해 파라미터를 변환하는 과정에서,

제1 계산 모듈(FireA) 및 제2 계산 모듈(FireB)을 포함하는 기본 계산 블록을 통해 연산을 수행하고,

제1 계산 모듈(FireA) 및 제2 계산 모듈(FireB) 각각은,

특징맵의 채널 수를 감소시키고, 이어지는 확장 계층의 연산 수를 감소시키기 위한 스퀴즈 계층; 및

채널을 다시 확장하기 위한 확장 계층을 포함하고,

수신 필드를 유지하여 정확도 저하를 최소화하기 위해 제1 계산 모듈의 스퀴즈 계층과 제2 계산 모듈의 스퀴즈 계층은 서로 다른 커널 크기를 갖는

FPGA 기반 CNN 가속기.

청구항 7

제6항에 있어서,

특징맵의 해상도를 절반으로 줄이고 최대 풀링 계층을 대체하기 위해 상기 기본 계산 블록의 스트라이드를 2로 설정하고,

CNN의 마지막에 추출된 특징을 분류하고 복수의 매개 변수를 포함하는 완전 연결 계층을 글로벌 평균 풀링으로 대체하고 커널 크기가 1인 컨볼루션 계층을 대체하여 상기 파라미터를 감소시키고,

CNN의 분류기는 컨볼루션 계층을 통해 클래스 수와 일치하도록 채널 수를 최소화한 후 특징맵의 해상도를 하나의 픽셀로 압축하며,

CNN의 수렴 안정성과 성능을 개선하기 위해 컨볼루션 계층과 활성화 함수 사이에 배치 정규화 계층을 삽입하는

FPGA 기반 CNN 가속기.

청구항 8

제5항에 있어서,

상기 CPU는,

플로팅-포인트 훈련을 통해 변환된 파라미터들에 대하여 양자화 인식 훈련을 통해 양자화된 파라미터를 추출하고, 정수-산술 전용(Integer-Arithmetic-Only) CNN 재구성을 수행하는 과정에서,

훈련 가능한 파라미터를 2의 거듭제곱 인수에 매핑하고, 첫 번째 에포크(epoch)의 첫 번째 배치의 활성화 데이터 통계 값으로 활성화 기준화 인수를 초기화하여 사전 보정 프로세스를 필요로 하지 않고, 기준화 인수를 2의 거듭제곱 항에 매핑함으로써 스케일에서 발생하는 곱셈 및 나눗셈 연산과 역-양자화 프로세스는 모두 이동 연산으로 대체되는 로그레벨 임계값 정량화(Log Level Threshold Quantization; LLTQ) 방법을 이용하는

FPGA 기반 CNN 가속기.

발명의 설명

기술 분야

[0001] 본 발명은 실시간 얼굴 감정 인식을 위한 자원 효율적인 정수-산술 전용 FPGA 기반 CNN 가속기 및 그 동작 방법에 관한 것이다.

배경 기술

[0002] 컴퓨터는 산업과 사회에서 중심적인 역할을 하고 있으며, 빠르게 일상 생활의 일부가 되고 있다. 이에 따라 인간과 컴퓨터 간의 상호작용에 대한 연구의 필요성이 증가하고 있다. 인간과 컴퓨터 간의 원활한 상호작용을 위해, 컴퓨터는 인간의 의도를 분석하고 그에 따라 반응할 수 있어야 한다. 표정에서 나타나는 감정은 인간의 의도를 표현하는 보편적이고 효과적인 방법이다. 얼굴에서 드러나는 표정을 통해 사람의 의도를 분석하는 것을 얼굴 표정 인식 기술이라고 하며, 자동차와 로봇 산업 등 다양한 분야에서 사용된다. 사람의 얼굴에서 나타나는 표정을 정확히 이해하기 위해서는 컴퓨터가 얼굴을 인식해 특정 표정군에 따라 자동으로 표정을 분류해야 한다.

[0003] 종래기술에서는 행복, 분노, 두려움, 놀라움, 혐오, 슬픔, 중립의 7가지 기본 감정을 정의하고, 기본적인 감정들이 인간의 문화와 상관없이 같은 방식으로 인식된다는 것을 확인할 수 있다. 컴퓨터 비전 분야에서는 종래기술에서 제안한 얼굴 표정 분석 방법인 FACS(Facial Action Coding System)이 대표적인 얼굴 감정 인식 모델로 사용되었다. FACS는 얼굴 근육의 기본적인 움직임을 AU(Action Units)으로 변형시킨 다음 얼굴 표정을 인식하기 위해 AU의 조합을 사용한다. 그러나, 실시간 자동화 시스템의 개발은 부적절한 계산 능력과 비효율적인 전처리 알고리즘으로 인한 문제가 있다.

[0004] 이미지 처리 알고리즘의 발달과 컴퓨팅 능력의 향상으로 얼굴 표정 인식 기술은 3상 파이프라인을 통해 수작업으로 만들어진 특징을 추출하는 방법으로 발전했다. 이 접근 방식은 이미지 전처리, 특징 추출 및 표현식 분류 단계로 구성된다. 이미지 전처리 단계는 필터링 및 히스토그램 균등화를 사용하여 관련 없는 정보를 제거하고 얼굴 표정과 관련된 정보를 강화한다. 특징 추출 단계는 가보 웨이블릿 커널(Gabor Wavelet Kernel), LBP(Local Binary Pattern), ASM(Active Shape Model), 하-라이크 피쳐 템플릿(Harr-Like Feature Template)과 같은 특징 추출기를 사용하여 이미지에서 얼굴 특징을 추출한다. 표현식 분류 단계는 서포트 벡터 머신(Support Vector Machine; SVM), KNN(K-Nearest Neighbor) 또는 AdaBoost 와 같은 분류기를 사용하여 추출된 특징을 표정군으로 분류한다. 핸드-크래프트(hand-crafted) 특징을 사용하는 방법은 적절한 특징 추출기와 표현식 분류기를 개별적으로 설계해야 하므로 두 단계를 동시에 최적화할 수 없다. 또한 외부 요인(예를 들어, 자세 변동, 패색, 조명)은 심각한 성능 저하를 야기할 수 있다.

[0005] 최근의 하드웨어 기술과 빅데이터 기술의 획기적인 발전으로 인해 딥러닝 기반 방법이 복잡하고 복잡한 문제에 탁월한 성능을 보이기 때문에 다양한 애플리케이션에 적용되고 있다. 특히 이미지 분류 및 물체 감지와 같은 컴퓨터 비전에서는 컨볼루션 신경망(Convolutional Neural Networks; CNN)을 사용하는 방법이 탁월한 성능을 보였다. 기존 방법과 달리 CNN 기반 접근 방식은 노이즈가 있는 환경에서 더 강력하여 이미지 전처리 및 특징 추출에 대한 의존도를 크게 줄인다. 또한 이 방법은 엔드 투 엔드 교육을 통해 파라미터를 한 번에 최적화할 수 있다. 얼굴 표정 인식 연구에서 CNN 기반 방법은 기존 방법보다 더 높은 성능을 보였지만 네트워크를 구축할 때 많은 컨볼루션 계층을 사용한다. 이와 같은 네트워크에는 많은 작업 및 메모리 풋프린트가 필요하다. 따라서,

첨단 운전자 지원 시스템(Advanced Driver Assistance System; ADAS)과 같이 실시간 처리가 필요한 애플리케이션에서 사용될 때는 제한적이다. 또한 가상현실(Virtual Reality; VR)과 같이 하드웨어 리소스가 제한된 환경에서는 적합하지 않다.

발명의 내용

해결하려는 과제

- [0006] 본 발명이 이루고자 하는 기술적 과제는 메모리 설치 공간 및 계산 복잡성을 감소시키기 위한 실시간 얼굴 감정 인식을 위한 자원 효율적인 정수-산술 전용 FPGA 기반 CNN 가속기 및 그 동작 방법을 제공하는데 있다. 또한, 본 발명의 실시예에 따른 CNN 가속기를 통해 정수 산술만을 사용하는 새로운 하드웨어 친화적인 양자화 방법을 제안하고, CNN의 일반화 및 분류 성능을 개선하기 위해 다양한 이미지 처리 알고리즘을 사용하여 만들어진 새로운 훈련 데이터셋인 FERPlus-A 데이터셋을 생성하여 훈련 후 양자화를 수행하는 방법 및 장치를 제공한다.

과제의 해결 수단

- [0007] 일 측면에 있어서, 본 발명에서 제안하는 CPU(Processing System; PS)와 FPGA(Programmable Logic, PL)를 포함하는 이기종 SoC(System on Chip) 플랫폼에서 구현되는 FPGA 기반 CNN 가속기의 동작 방법은 CPU에서 CNN 얼굴 표정 인식 성능을 향상시키기 위해 복수의 이미지 처리 알고리즘을 결합한 훈련 데이터셋을 생성하고, 생성된 훈련 데이터셋에 대하여 플로팅-포인트(floating-point) 훈련을 통해 파라미터를 변환하는 단계, CPU에서 플로팅-포인트 훈련을 통해 변환된 파라미터들에 대하여 양자화 인식 훈련을 통해 양자화된 파라미터를 추출하고, 정수-산술 전용(Integer-Arithmetic-Only) CNN 재구성을 수행하는 단계 및 이기종 SoC 플랫폼의 FPGA 및 CPU 영역에서 상기 정수-산술 전용 CNN을 이용하여 실시간 얼굴 감정 인식을 수행하는 단계를 포함한다.
- [0008] 상기 CPU에서 CNN 얼굴 표정 인식 성능을 향상시키기 위해 복수의 이미지 처리 알고리즘을 결합한 훈련 데이터셋을 생성하고, 생성된 훈련 데이터셋에 대하여 플로팅-포인트 훈련을 통해 파라미터를 변환하는 단계는 제1 계산 모듈(FireA) 및 제2 계산 모듈(FireB)을 포함하는 기본 계산 블록을 통해 연산을 수행하고, 제1 계산 모듈(FireA) 및 제2 계산 모듈(FireB) 각각은, 특징맵의 채널 수를 감소시키고, 이어지는 확장 계층의 연산 수를 감소시키기 위한 스퀴즈 계층 및 채널을 다시 확장하기 위한 확장 계층을 포함하고, 수신 필드를 유지하여 정확도 저하를 최소화하기 위해 제1 계산 모듈의 스퀴즈 계층과 제2 계산 모듈의 스퀴즈 계층은 서로 다른 커널 크기를 갖는다.
- [0009] 특징맵의 해상도를 절반으로 줄이고 최대 풀링 계층을 대체하기 위해 상기 기본 계산 블록의 스트라이드를 2로 설정하고, CNN의 마지막에 추출된 특징을 분류하고 복수의 매개 변수를 포함하는 완전 연결 계층을 글로벌 평균 풀링으로 대체하고 커널 크기가 1인 컨볼루션 계층을 대체하여 상기 파라미터를 감소시키고, CNN의 분류기는 컨볼루션 계층을 통해 클래스 수와 일치하도록 채널 수를 최소화한 후 특징맵의 해상도를 하나의 픽셀로 압축하며, CNN의 수렴 안정성과 성능을 개선하기 위해 컨볼루션 계층과 활성화 함수 사이에 배치 정규화 계층을 삽입한다.
- [0010] 상기 CPU에서 플로팅-포인트 훈련을 통해 변환된 파라미터들에 대하여 양자화 인식 훈련을 통해 양자화된 파라미터를 추출하고, 정수-산술 전용 CNN 재구성을 수행하는 단계는 훈련 가능한 파라미터를 2의 거듭제곱 인수에 매핑하고, 첫 번째 에포크(epoch)의 첫 번째 배치의 활성화 데이터 통계 값으로 활성화 기준화 인수를 초기화하여 사전 보정 프로세스를 필요로 하지 않고, 기준화 인수를 2의 거듭제곱 항에 매핑함으로써 스케일에서 발생하는 곱셈 및 나눗셈 연산과 역-양자화 프로세스는 모두 이동 연산으로 대체되는 로그레벨 임계값 정량화(Log Level Threshold Quantization; LLTQ) 방법을 이용한다.
- [0011] 또 다른 일 측면에 있어서, 본 발명에서 제안하는 CPU(Processing System; PS)와 FPGA(Programmable Logic, PL)를 포함하는 이기종 SoC(System on Chip) 플랫폼에서 구현되는 FPGA 기반 CNN 가속기에 있어서, 상기 CPU는 CNN 얼굴 표정 인식 성능을 향상시키기 위해 복수의 이미지 처리 알고리즘을 결합한 훈련 데이터셋을 생성하고, 생성된 훈련 데이터셋에 대하여 플로팅-포인트(floating-point) 훈련을 통해 파라미터를 변환하고, 플로팅-포인트 훈련을 통해 변환된 파라미터들에 대하여 양자화 인식 훈련을 통해 양자화된 파라미터를 추출하고, 정수-산술 전용(Integer-Arithmetic-Only) CNN 재구성을 수행하며, 이기종 SoC 플랫폼의 FPGA 및 CPU 영역에서 상기 정수-산술 전용 CNN을 이용하여 실시간 얼굴 감정 인식을 수행한다.

발명의 효과

[0012] 본 발명의 실시예들에 따른 실시간 얼굴 감정 인식을 위한 자원 효율적인 정수-산술 전용 FPGA 기반 CNN 가속기 및 그 동작 방법을 통해 메모리 설치 공간 및 계산 복잡성을 감소시킬 수 있다. 또한, 본 발명의 실시예에 따른 CNN 가속기를 통해 정수 산술만을 사용하는 새로운 하드웨어 친화적인 양자화 방법을 제안하고, 다양한 이미지 처리 알고리즘을 사용하여 만들어진 새로운 훈련 데이터셋인 FERPlus-A 데이터셋을 생성하여 훈련 후 양자화를 수행함으로써 CNN의 일반화 및 분류 성능을 개선할 수 있다.

도면의 간단한 설명

[0013] 도 1은 본 발명의 일 실시예에 따른 실시간 얼굴 감정 인식을 위한 자원 효율적인 정수-산술 전용 FPGA 기반 CNN 가속기의 동작 방법을 설명하기 위한 도면이다.

도 2는 본 발명의 일 실시예에 따른 경량 CNN 아키텍처의 메모리 사용 및 계산 복잡성을 최적화하는 과정을 설명하기 위한 도면이다.

도 3은 본 발명의 일 실시예에 따른 양자화 방법 LLTQ를 이용한 균일 양자화 과정과 종래기술의 LSQ 및 TQT 양자화 과정을 설명하기 위한 도면이다.

도 4는 본 발명의 일 실시예에 따른 정수 스케일 변환 방법을 설명하기 위한 도면이다.

도 5는 본 발명의 일 실시예에 따른 실시간 얼굴 감정 인식을 위한 자원 효율적인 정수-산술 전용 FPGA 기반 CNN 가속기의 아키텍처를 나타낸다.

도 6은 본 발명의 일 실시예에 따른 정수-산술 전용 FPGA 기반 CNN 가속기에 대한 전체 작업 스케줄링 과정을 설명하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0014] 최근 컴퓨터 비전에서 탁월한 성능을 보이는 컨볼루션 신경망(Convolutional Neural Networks; CNN)을 이용한 얼굴 표정 인식에 대한 많은 연구가 수행되고 있다. 높은 분류 정확도를 얻으려면 파라미터가 많고 계산 복잡성이 높은 CNN 아키텍처가 필요하다. 그러나 하드웨어 리소스가 제한된 임베디드 시스템에는 적합하지 않다.

[0015] 본 발명에서는 임베디드 시스템에 최적화된 경량 CNN 아키텍처를 제안한다. 제안된 CNN 아키텍처는 메모리 설치 공간이 작고 계산 복잡성이 낮다. 또한, 정수 산술만을 사용하는 새로운 하드웨어 친화적인 양자화 방법을 제안한다. 제안된 하드웨어 친화 양자화 방법은 기증화 인수를 2의 거듭제곱 항으로 매핑하고 기증화 인수를 사용한 곱셈과 나눗셈 연산을 이동 연산으로 대체한다.

[0016] 또한, CNN의 일반화 및 분류 성능을 개선하기 위해 FERPlus-A 데이터셋을 생성한다. 이것은 다양한 이미지 처리 알고리즘을 사용하여 만들어진 새로운 훈련 데이터셋이다. FERPlus-A로 FERPlus-A 데이터셋을 훈련한 후 양자화를 수행한다.

[0017] 본 발명의 실시예에 따라 양자화된 CNN 파라미터의 크기는 약 0.39 MB이며, 작동 횟수는 약 28M 정수 연산(Integer Operations; IOP)이다. FERPlus 테스트 데이터셋에서 정수 산술만을 사용하여 양자화된 CNN의 성능을 평가했고 분류 정확도는 약 86.58%로서 종래기술의 다른 경량 CNN보다 높은 정확도를 달성했다. 정수 산술만 사용하는 제안된 CNN 아키텍처는 Xilinx ZC706 SoC 플랫폼에서 병렬화 전략과 효율적인 데이터 캐싱 전략을 적용하여 실시간 얼굴 표정 인식을 위해 구현된다. 실시간 얼굴 표정 인식을 위해 구현된 FPGA 기반 CNN 가속기는 250MHz에서 대략 초당 10 프레임(Frame per Second; FPS)을 달성하고 2.3W를 소비한다.

[0018] 경량 CNN을 사용한 여러 얼굴 표정 인식 기술에 있어서, 종래기술에서는 클라우드 소싱 및 레이블 분배 방법을 사용하여 FER2013 데이터셋의 문제를 해결한 FERPlus 데이터셋을 구성했으며, 가벼운 VGG를 사용한 네 가지 다른 훈련 기술이 제안되었다. 또 다른 종래기술에서는 DeepID2 와 ResNet 을 통해 식별 특징과 표정 특징을 각각 추출했으며, 각 CNN 출력의 특징을 결합하여 분류했다. 또 다른 종래기술에서는 사전 훈련된 파라미터를 이미지 분류에 자주 사용되는 데이터셋과 함께 사용하는 전층 학습 및 앙상블 방법을 제안했다. 또 다른 종래기술에서는 원래 이미지 데이터셋의 소벨 경사도(Sobel gradient)와 라플라시안(Laplacian)을 통합한 데이터셋으로 MobileNetv2를 훈련시키는 접근 방식을 제안했다. 또 다른 종래기술에서는 CNN의 변별력을 개선하기 위해 글로벌(global) 손실에 센터(center) 손실을 추가하는 훈련 방법을 사용했다.

[0019] 또 다른 종래기술에서는 경사도 가중 클래스 활성화 매핑(Gradient Weighted Class Activation Mapping; Grad-CAM)을 사용하여 두 개의 사용자 정의 CNN을 구성했다. 그들은 또한 두 개의 CNN에서 추출한 특징들을 결합하여

얼굴 표정 인식에 사용하는 이중 통합 CNN을 제안했다. 또 다른 종래기술에서는 SHCNN을 제안했으며, 이는 비교적 작은 데이터셋의 과적합 문제를 완화하고 정적 및 마이크로 표현을 동시에 인식할 수 있다. 또 다른 종래기술에서는 DenseNet을 통해 추출된 특징에서 클래스 활성화 매핑(Class Activation Mapping; CAM)을 사용하여 다양한 얼굴 영역의 기여를 양자화하는 방법을 도입했다. 또 다른 종래기술에서는 두 개의 유사한 표정군을 하나로 통합하는 새로운 데이터셋을 만들었다. 여기서 얼굴 감지와 얼굴 정렬 방법을 적용하여 DenseNet을 훈련시키는 방법을 제안했다.

[0020] 그러나 또 다른 종래기술들은 많은 양의 메모리 공간을 차지하고 높은 계산 복잡성의 문제를 가지고 있었다. 일부 종래기술들에서 CNN은 더 작은 파라미터로 사용되었지만 두 개의 CNN이 추론을 위해 사용되었기 때문에 여전히 많은 양의 메모리 공간을 차지했으며 실시간 처리에 적합하지 않은 상태로 남아 있다. 일부 종래기술들에서 CNN은 파라미터가 작고 컴퓨팅 복잡성이 낮지만 얼굴 표정 인식 정확도는 매우 낮다. 소수의 파라미터를 사용하는 종래기술에서도 조밀한 연결로 인해 너무 많은 메모리 공간을 요구하기 때문에 실시간 처리에 적합하지 않았다. 이것이 DenseNet의 특징이다. 이러한 문제를 해결하기 위해서는 얼굴 표정 인식의 정확성을 유지하면서 메모리 풋프린트와 계산 복잡성을 최적화하여 경량 CNN 아키텍처를 설계하기 위한 기술을 필요로 한다.

[0021] 양자화는 32비트 플로팅-포인트(Floating-Point; FP)으로 훈련된 파라미터를 저비트 고정 소수점(Fixed-Point; FX) 또는 정수(Integer; INT)로 변환하는 방법이다. 양자화를 적용하면 메모리 사용을 최적화하고 간단한 하드웨어를 사용하여 계산 횟수를 줄일 수 있다. 양자화는 두 가지 방법으로 나뉜다. 훈련 후 양자화(Post-Training Quantization; PTQ) 방법은 사전 훈련된 파라미터를 사용하여 회의할 때 보정을 적용하여 양자화 오류를 최소화한다. 양자화 인식 훈련(Quantization-Aware Training; QAT) 방법은 양자화의 효과를 고려하여 파라미터를 재훈련한다.

[0022] 종래기술에서는 양자화 과정 중 편향된 가중치 오류와 불균형을 해결하기 위해 편향 보정 및 교차 계층 평준화 방법을 제안했다. 또 다른 종래기술에서는 플로팅-포인트 값과 양자화 오차 사이의 최소 평균 제곱 오차를 사용하여 양자화 오류를 최소화하는 방법을 도입했다. 또 다른 종래기술에서는 정수 양자화 분석 클리핑(Analytical Clipping for Integer Quantization; ACIQ)을 사용하여 최적의 클리핑 값을 계산하여 양자화 오류를 최소화하였다. 또 다른 종래기술에서는 최적의 양자화 범위를 얻기 위해 배치 정규화(Batch Normalization; BN) 계층의 평균과 분산을 이용한 증류 데이터셋을 제안했다.

[0023] PTQ 방법은 재훈련 과정이 필요하지 않기 때문에 컴퓨팅 리소스와 최적화 시간을 절약하고 신속한 배치를 가능하게 한다. 단, 편향 보정 및 교차 계층 평준화 방법 및 플로팅-포인트 값과 양자화 오차 사이의 최소 평균 제곱 오차를 사용하여 양자화 오류를 최소화하는 방법에서는 정확도 손실을 최소화하기 위해 미세 조정 값이 필요하다. ACIQ 방법 및 BN 방법에서는 정확도 저하를 최소화하기 위해 각 채널에 대해 서로 다른 비트 수를 할당해야 한다. 다른 종래기술에서는 정확도를 유지하기 위해 각 채널에 대한 별도의 기준화 인수를 구해야 한다. 또한, 모든 PTQ 방법은 플로팅-포인트 파라미터를 요구하기 때문에 하드웨어를 구현할 때 전용 연산자가 필요하다.

[0024] 또 다른 종래기술에서는 적절한 기준화 인수를 찾기 위한 파라미터화된 클리핑 활성화(Parameterized Clipping Activation; PACT) 함수를 정의하여 양자화 오류를 최소화하는 방법을 제안했다. 또 다른 종래기술에서는 정수-산술만을 사용하여 행렬 곱셈을 수행하는 양자화기의 작동을 정의하고 계층 융합 방법을 제시하였다. 또 다른 종래기술에서는 파라미터화된 양자화 간격을 사용하여 작업 손실을 최소화하는 방법을 제안했다. 또 다른 종래기술에서는 기준화 인수를 학습 가능한 파라미터로 만들어 최적의 양자화 매핑을 학습하는 방법을 제안했다. 또 다른 종래기술에서는 기준화 인수 변환 방정식을 사용하여 기준화 인수를 2의 거듭제곱 항에 매핑하는 방법을 제안했다.

[0025] QAT 방법은 다른 파라미터로 양자화 파라미터를 훈련시킴으로써 PTQ 방법보다 높은 정확도를 얻을 수 있다. 그러나 일부 종래기술에서는 BN 계층을 양자화하지 않았다. 일부 종래기술에서는 양자화에 필요한 파라미터는 플로팅-포인트며 하드웨어를 구현할 때 전용 연산자가 필요하다. 다른 접근법에서는 QAT를 시작하기 전에 활성화 기준화 인수의 보정이 필요하다. 이러한 문제를 보완하기 위해서는 사전 보정 단계가 필요하지 않은 하드웨어 친화적인 양자화 방법이 필요하다.

[0026] GPU를 사용하여 얼굴 표정 인식을 수행하는 것은 처리량과 속도 면에서 최고의 성능을 제공한다. 그러나 임베디드 시스템에서 GPU를 사용하는 것은 어렵다. 이러한 시스템은 에너지 소비와 하드웨어 리소스 사용을 줄이는 솔루션을 필요로 한다. 필드 프로그램 가능 게이트 어레이(Field Programmable Gate Array; FPGA)는 설계자가 애플리케이션에 맞게 설계를 조정할 수 있는 가능성을 가지고 있다. 최적의 작동 속도와 합리적인 전력 소비를 가

능하게 프로그래밍할 수 있기 때문에 임베디드 시스템의 실시간 처리에 적합하다.

- [0027] 이러한 장점 때문에 얼굴 표정 인식을 위한 여러 FPGA 기반 CNN 가속기가 제안되었다. 종래기술에서는 높은 수준의 합성(High Level-Synthesis; HLS)을 사용하여 Xilinx Zynq-XC7Z020 FPGA에 CNN 가속기를 구현했다.
- [0028] 블록램(Block RAM; BRAM) 사용량을 줄이기 위해 D램에 계층의 결과와 입력을 저장하는 CNN 가속기를 설계하고 고속 D램 접속을 위해 VDMA를 사용했다. 또 다른 종래기술에서는 얼굴 감정 인식을 위한 효율적인 구조와 데이터 전처리 방법을 적용하여 CNN 가속기를 설계하고 Verilog HDL을 사용하여 Altera DE-10 FPGA에 구현했다. 또 다른 종래기술에서 가속기는 Altera DE-10 FPGA의 처리 엔진 코어를 구성하여 컨볼루션 작동을 가속화하였다. 또 다른 종래기술에서는 CNN을 구성하는 다양한 컨볼루션 계층에 대해 서로 다른 병렬 처리를 가진 Verilog HDL을 사용하는 Altera Cyclone-V FPGA 가속기를 제안했다. 또 다른 종래기술에서 CNN 가속기는 Verilog HDL을 사용하여 Altera Cyclone-V FPGA에 구현되었다. 그들은 FPGA에서 DSP 사용을 최대화하기 위해 구성 가능한 컨볼루션 컴퓨팅 어레이를 설계했다.
- [0029] 일부 종래기술에서는 완전 연결, 컨볼루션 및 풀링 계층을 포함하는 간단한 CNN 아키텍처를 구현했다. 파라미터의 수와 계산 복잡성은 낮았지만 정확성도 낮아서 매우 정확한 얼굴 표정 인식이 필요한 애플리케이션에는 적합하지 않았다. 일부 종래기술에서 DeepID 아키텍처는 많은 하드웨어 리소스를 필요로 하는 FPGA에서 구현되어 임베디드 시스템에서는 사용할 수 없다. 따라서 실시간 얼굴 표정 인식을 위한 저전력 저비용 FPGA 기반 CNN 가속기를 설계하고 높은 얼굴 표정 인식 성능을 유지해야 한다.
- [0031] 도 1은 본 발명의 일 실시예에 따른 실시간 얼굴 감정 인식을 위한 자원 효율적인 정수-산술 전용 FPGA 기반 CNN 가속기의 동작 방법을 설명하기 위한 도면이다.
- [0032] 실시간 얼굴 감정 인식을 위한 자원 효율적인 정수-산술 전용 FPGA 기반 CNN 가속기의 동작 방법은 얼굴 표정 인식의 정확성을 유지하면서 메모리 풋프린트와 계산 복잡성을 최적화함으로써 달성된다. 또한, 종래기술의 단점을 해결하기 위한 로그레벨 임계값 정량화(Log Level Threshold Quantization; LLTQ)가 제안된다[1], [2]. 또한 실시간 얼굴 표정 인식을 위해 저전력 저비용 FPGA 기반 CNN 가속기가 제안된다.
- [0033] 제안하는 CPU(Processing System; PS)와 FPGA(Programmable Logic, PL)를 포함하는 이기종 SoC 플랫폼에서 구현되는 FPGA 기반 CNN 가속기의 동작 방법은 CPU에서 CNN 얼굴 표정 인식 성능을 향상시키기 위해 복수의 이미지 처리 알고리즘을 결합한 훈련 데이터셋을 생성하고, 생성된 훈련 데이터셋에 대하여 플로팅-포인트(floating-point) 훈련을 통해 파라미터를 변환하는 단계(110), CPU에서 플로팅-포인트 훈련을 통해 변환된 파라미터들에 대하여 양자화 인식 훈련을 통해 양자화된 파라미터를 추출하고, 정수-산술 전용(Integer-Arithmetic-Only) CNN 재구성을 수행하는 단계(120) 및 이기종 SoC(System on Chip) 플랫폼의 FPGA(Programmable Logic; PL) 및 CPU(Processing System; PS) 영역에서 상기 정수-산술 전용 CNN을 이용하여 실시간 얼굴 감정 인식을 수행하는 단계(130)를 포함한다.
- [0034] 단계(110)에서, CPU에서는 CNN 얼굴 표정 인식 성능을 향상시키기 위해 복수의 이미지 처리 알고리즘을 결합한 훈련 데이터셋을 생성하고(111), 본 발명의 실시예에 따른 새로운 경량 CNN 아키텍처(112)를 이용하여, 상기 생성된 훈련 데이터셋에 대한 플로팅-포인트 훈련을 통해 파라미터를 변환하고(113), PTH File을 생성한다(114). 본 발명의 실시예에 따른 PTH File은 플로팅-포인트 훈련된 파라미터들이다.
- [0035] 단계(110)에서, 제1 계산 모듈(FireA) 및 제2 계산 모듈(FireB)을 포함하는 기본 계산 블록을 통해 연산을 수행할 수 있다. 제1 계산 모듈(FireA) 및 제2 계산 모듈(FireB) 각각은, 특징맵의 채널 수를 감소시키고, 이어지는 확장 계층의 연산 수를 감소시키기 위한 스퀴즈 계층 및 채널을 다시 확장하기 위한 확장 계층을 포함한다. 이때, 수신 필드를 유지하여 정확도 저하를 최소화하기 위해 제1 계산 모듈의 스퀴즈 계층과 제2 계산 모듈의 스퀴즈 계층은 서로 다른 커널 크기를 갖는다.
- [0036] 본 발명의 실시예에 따르면, 특징맵의 해상도를 절반으로 줄이고 최대 풀링 계층을 대체하기 위해 상기 기본 계산 블록의 스트라이드를 2로 설정할 수 있다. 또한, CNN의 마지막에 추출된 특징을 분류하고 복수의 매개 변수를 포함하는 완전 연결 계층을 글로벌 평균 풀링으로 대체하고 커널 크기가 1인 컨볼루션 계층을 대체하여 상기 파라미터를 감소시킬 수 있다.
- [0037] 본 발명의 실시예에 따르면, CNN의 분류기는 컨볼루션 계층을 통해 클래스 수와 일치하도록 채널 수를 최소화한 후 특징맵의 해상도를 하나의 픽셀로 압축하며, CNN의 수렴 안정성과 성능을 개선하기 위해 컨볼루션 계층과 활

성화 함수 사이에 배치 정규화 계층을 삽입할 수 있다.

- [0038] 단계(120)에서, CPU에서 플로팅-포인트 훈련을 통해 변환된 파라미터들에 대하여 양자화 인식 훈련(121)을 통해 양자화된 파라미터를 추출하고(122), 정수-산술 전용(Integer-Arithmetic-Only) CNN 재구성을 수행하여(123), BIN File을 생성한다(124). 본 발명의 실시예에 따른 BIN File은 직렬화된 정수 파라미터들이다.
- [0039] 본 발명의 실시예에 따른 로그레벨 임계값 정량화(LLTQ) 방법은 훈련 가능한 파라미터를 2의 거듭제곱 인수에 매핑하고, 첫 번째 에포크(epoch)의 첫 번째 배치의 활성화 데이터 통계 값으로 활성화 기준화 인수를 초기화하여 사전 보정 프로세스를 필요로 하지 않는다. 또한, 기준화 인수를 2의 거듭제곱 항에 매핑함으로써 스케일에서 발생하는 곱셈 및 나눗셈 연산과 역-양자화 프로세스는 모두 이동 연산으로 대체된다.
- [0040] 단계(130)에서는, 이기종 SoC(System on Chip) 플랫폼의 FPGA(Programmable Logic; PL) 영역(131) 및 CPU(Processing System; PS) 영역(132)에서 상기 정수-산술 전용 CNN을 이용하여 실시간 얼굴 감정 인식을 수행한다.
- [0041] 본 발명의 실시예에 따른 FPGA 영역(131)에서는 XSA File(143)을 이용할 수 있다. XSA(Xilinx support Ardhive) File 생성 과정(140)은 HLS 컴파일러(141) 및 CNN 가속기 로직(142)을 거쳐 생성될 수 있다.
- [0042] 본 발명의 실시예에 따른 CPU 영역(132)에서는 ELF File(153)을 이용할 수 있다. ELF(Executable and linkable) File 생성 과정(150)은 표준 C/C++ 컴파일러(151) 및 장치 드라이버(152)을 거쳐 생성될 수 있다.
- [0043] 이와 같이, 본 발명의 실시예에 따른 CNN 가속기 로직(142) 및 장치 드라이버(152)는 실시간 얼굴 표정 인식 애플리케이션을 위해 이기종 SoC 플랫폼의 PL 및 PS 영역에 구현되었다.
- [0045] 도 2는 본 발명의 일 실시예에 따른 경량 CNN 아키텍처의 메모리 사용 및 계산 복잡성을 최적화하는 과정을 설명하기 위한 도면이다.
- [0046] 도 2를 참조하면, 21개의 컨볼루션 계층과 하나의 글로벌 평균 풀링 계층으로 구성된 본 발명의 실시예에 따른 경량 CNN 아키텍처를 나타낸다. 계산 블록에는 두 개의 기본 계산 모듈인 FireA와 FireB가 포함되어 있다. 다음 계산 블록의 파라미터인 스퀴즈 계층(s)의 필터 수와 확장 계층(e)의 필터 수는 다음 계산 블록에서 두 배 증가한다. 본 발명의 실시예에 따른 CNN은 다음과 같은 기법을 적용하여 설계되었다: 기본 연산 모듈, 모든 컨볼루션 및 정규 네트워크 및 2의 거듭제곱 해상도 및 4 배수 출력 채널.
- [0047] 먼저, 본 발명의 실시예에 따른 기본 연산 모듈에 대하여 설명한다.
- [0048] 본 발명의 실시예에 따른 경량 CNN은 스퀴즈넷(SqueezeNet)에 제안된 fire 모듈을 응용한다. fire 모듈은 두 가지 유형의 계층으로 구성된다. 스퀴즈 계층(Squeeze Layer; SQ)은 특징맵의 채널 수를 줄이고, 이어지는 확장 계층의 연산 수를 줄일 수 있다. 확장 계층은 채널을 다시 확장하는 역할을 하며 커널 크기가 3인 계층과 커널 크기가 1인 계층 두 가지 계층으로 구성된다. SQ의 출력 특징맵은 두 개의 경로로 나뉘며 각각 EX1과 EX3으로 입력된다. 또한 EX1과 EX3의 출력 특징맵은 채널별 연결을 통해 병합된 후 다음 계층으로 들어간다.
- [0049] 본 발명의 실시예에 따른 경량 CNN은 기본 계산 모듈로 수정된 fire 모듈을 사용하며 두 가지 유형(FireA 및 FireB)으로 구성된다. FireA와 FireB의 확장 계층 구성은 동일하다. 그러나 SQ의 커널 크기는 FireA에서 3으로, FireB에서 1로 설정된다. FireA에서 커널 크기를 3으로 설정하는 이유는 적절한 수신 필드를 유지하여 정확도 저하를 최소화하기 위함이다.
- [0050] 다음으로, 본 발명의 실시예에 따른 모든 컨볼루션 및 정규 네트워크에 대하여 설명한다.
- [0051] 본 발명의 실시예에 따른 경량 CNN은 커널 크기가 3 또는 1인 컨볼루션 계층만 사용하므로 하드웨어에서 구현할 때 다른 커널 크기에 대해서는 가속 방법을 고려할 필요가 없다. 최대 풀링 계층은 특징맵의 해상도를 절반으로 줄이고 정보를 압축한다. 특징맵의 최대값을 찾아야 하기 때문에 하드웨어에서 구현될 때 병목 현상이 발생할 수 있다. 제안하는 CNN은 FireA의 스트라이드를 2로 설정한다. 이는 특징맵의 해상도를 절반으로 줄이고 최대 풀링 계층을 대체한다.
- [0052] 본 발명의 실시예에 따른 완전 연결(Fully-Connected; FC) 계층은 CNN의 맨 마지막에 추출된 특징을 분류하며 많은 매개 변수를 포함한다. 파라미터의 수는 FC 계층을 글로벌 평균 풀링으로 대체하고 커널 크기가 1인 컨볼루션 계층을 대체하여 줄일 수 있다. 구성된 분류기는 컨볼루션 계층을 통해 클래스 수와 일치하도록 채널 수를

최소화한 다음 특징맵의 해상도를 한 픽셀로 압축한다. 모든 계층의 활성화 함수는 ReLU라는 레이블이 붙어 있다. 하드웨어에서 구현될 때 ReLU 함수는 부호 비트 비교만 사용하기 때문에 매우 효율적이다. 이와 같은 방법으로 CNN의 모든 계층은 컨볼루션 계층으로 만들어졌고 매우 일반적인 네트워크로 설계되었다. 또한 CNN 수렴 안정성과 성능을 개선하기 위해 컨볼루션 계층과 활성화 함수 사이에 BN 계층을 삽입했다. BN 계층은 BN 계층 융합을 통해 컨볼루션 계층과 결합될 수 있기 때문에, 하드웨어에서 구현할 때 BN 계층에 의해 생성된 추가 연산과 파라미터의 수는 무시될 수 있다.

[0053] 다음으로, 본 발명의 실시예에 따른 2의 거듭제곱 해상도 및 4 배수 출력 채널에 대하여 설명한다.

[0054] 본 발명의 실시예에 따르면, 특징맵의 너비와 높이를 모두 2의 거듭제곱 항으로 만듦으로써 특징맵의 주소를 계산하는 부분의 곱셈과 나눗셈 연산을 이동 연산으로 대체할 수 있다. 또한, 컨볼루션 계층의 출력 채널은 모두 효율적 병렬 다중 누적(Multiply-Accumulate; MAC) 하드웨어 작동을 위해 4의 배수로 설정되었다.

[0056] 도 3은 본 발명의 일 실시예에 따른 양자화 방법 LLTQ를 이용한 균일 양자화 과정과 종래기술의 LSQ 및 TQT 양자화 과정을 설명하기 위한 도면이다.

[0057] 본 발명의 실시예에 따른 로그레벨 임계값 양자화(LLTQ) 기법에 관하여 설명한다.

[0058] 학습된 단계 크기 양자화(Learned Step size Quantization; LSQ)방법[1]과 훈련된 양자화 임계값(Trained Quantization Threshold; TQT) 방법[2]은 균일한 양자화를 사용한다. 균일한 양자화기가 삽입된 CNN은 사전 훈련된 플로팅-포인트 파라미터를 사용하여 양자화 인식 훈련(Quantization-Aware Training; QAT)을 수행한다.

[0059] 도 3은 균일한 양자화기(320)가 삽입된 컨볼루션 계층(330)의 구조와 양자화기(310)의 4단계 작동을 보여준다. 첫 번째 단계인 스케일 프로세스, 양자화기는 실제 값 범위를 지정된 비트의 정수 값 범위에 매핑한다. 라운드 프로세스라고 하는 두 번째 단계는 정수 값 범위로 매핑된 모든 실수 값을 정수로 변환한다. 세 번째 단계는 클램프 프로세스이다. 이 프로세스는 양자화 수준에 맞게 양자화 범위를 초과하는 요소를 제거한다. 양자화 비트가 n인 경우, 클램프 프로세스 후의 값은 (1)에 표시된 바와 같이 바운드(Bound) 내부에 위치한다.

$$Bound = \begin{cases} [-2^{n-1}, 2^{n-1} - 1], & \text{signed data} \\ [0, 2^n - 1], & \text{unsigned data.} \end{cases} \quad (1)$$

[0061] 네 번째 단계인 역-양자화(de-quant) 프로세스는 정수 범위에 매핑된 값을 양자화의 영향을 받는 플로팅-포인트 값으로 변환한다. 양자화된 값을 정수로 다시 플로팅-포인트로 변환하는 이유는 GPU를 사용하는 대부분의 기존 딥러닝 프레임워크가 플로팅-포인트 훈련에 최적화되어 있기 때문이다. 순방향 경로에 대한 전체 식은 다음과 같이 설명할 수 있다.

$$x_d = clamp \left(round \left(\frac{x_f}{s} \right), Bound \right) \cdot s \quad (2)$$

[0063] 여기서 x_f 는 양자화기의 입력인 플로팅-포인트 값이고 x_d 는 역-양자화 프로세스를 통과한 양자화기의 출력 값이다. $clamp(x, Bound)$ 함수는 균일 양자화기의 두 번째 단계이며 반올림된 값(x)을 경계 내에 맞도록 자른다. $round(\cdot)$ 함수는 균일 양자화기의 세 번째 단계이다. 실제 데이터를 가장 가까운 정수로 반올림한다. STE(Straight-Through Estimator)[8]를 사용하여 반올림 공정에서 발생하는 불연속성을 해결하였다.

[0064] LSQ [1] 방법은 훈련 가능한 파라미터를 기준화 인수로 설정하고 다른 파라미터와 함께 기준화 인수를 훈련시킨다. 따라서 LSQ는 양자화 공정에서 기준화 인수를 변환할 필요가 없다. 이것은 제안하는 경량 CNN에 사용하기 위해 잘 알려져 있지만 하드웨어에 이를 구현할 때 전용 연산자가 필요하다는 단점이 있다. 이는 플로팅-포인트 기준화 인수이기 때문이다. TQT [2] 방법은 훈련 가능한 파라미터를 로그 임계값($\log_2 t$)으로 설정하고 기준화 인수를 다음과 같이 2의 거듭제곱 항에 매핑한다.

$$s = \frac{2^{ceil(\log_2 t)}}{q_level} \quad (3)$$

[0066] 여기서 q_level 은 식(1)의 Bound 최대값에 1을 더하는 값이다.

[0067] TQT는 기준화 인수를 2의 거듭제곱 항에 매핑하기 때문에 기준화 인수를 사용한 곱셈과 나눗셈을 이동 연산으로 대체할 수 있다는 장점이 있다. 단, 재훈련을 시작하기 전에 활성화 기준화 인수의 보정이 필요하다. 따라서 본 발명에서는 이전 연구 LSQ, TQT의 문제를 해결하는 새로운 하드웨어 친화적 양자화 방법을 제안한다.

[0068] 제안하는 LLTQ는 훈련 가능한 파라미터를 직접 2의 거듭제곱 인수에 매핑할 수 있다. 또한 TQT와 달리 LLTQ 방법을 사용하면 첫 번째 에포크(epoch)의 첫 번째 배치의 활성화 데이터 통계 값으로 활성화 기준화 인수를 초기화할 수 있으므로 사전 보정 프로세스가 필요하지 않다. 또한 기준화 인수를 2의 거듭제곱 항에 매핑함으로써 스케일에서 발생하는 곱셈 및 나눗셈 연산과 역-양자화 프로세스는 모두 이동 연산으로 대체될 수 있다.

[0069] 본 발명의 실시예에 따른 LLTQ 방법의 성능은 정수 스케일 변환 방법을 통한 정수 산술만을 사용하여 평가된다. LLTQ는 훈련 가능한 파라미터를 레벨 임계값(level_th)으로 설정하고 다음 식을 사용하여 기준화 인수를 2의 거듭제곱 항에 매핑한다.

$$s = \frac{2^{\text{ceil}(\log_2(\text{abs}(\text{level_th} * q_level)))}}{q_level} \quad (4)$$

[0070] 여기서 q_level은 식(3)에서와 같다. 기준화 인수를 계산하려면 많은 연산이 필요하다. 그러나 정수 산술만 사용하도록 재구성된 CNN을 사용하여 추론을 수행할 경우 2의 거듭제곱 항에 매핑된 기준화 인수의 이동량만 필요하다.

[0071] 또한, LLTQ에 대하여 하드웨어 친화적인 다양한 양자화 방식이 적용되었다. 대칭 양자화를 사용함으로써 균일한 양자화 중에 발생하는 영점에 대한 계산 오버헤드가 제거된다. 계층별 스케일링을 도입함으로써 주어진 파라미터 또는 입력 텐서의 모든 요소에 대해 하나의 기준화 인수로 양자화를 수행할 수 있다. 컨볼루션 층과 BN 계층의 파라미터를 결합한 BN 계층 융합이 적용되었다. 이 방식은 BN 계층에서 양자화를 진행할 필요가 없다. 이는 BN 계층에 필요한 곱셈과 나눗셈 연산을 제거함으로써 연산의 수와 추론 시간을 줄일 수 있다. fire 모듈은 스쿼즈 계층의 출력이 두 확장 계층의 입력으로 들어가는 구조이다. 플로팅-포인트에서는 두 확장 계층의 입력이 동일하다. 그러나 QAT에서 각 컨볼루션 계층에는 서로 다른 기준화 인수가 필요하다. 따라서 스쿼즈 계층은 각 확장 계층의 출력을 별도로 계산해야 한다. 이는 하드웨어에서 CNN 추론을 수행할 때 병목 현상을 일으킬 수 있다. 이 문제를 해결하기 위해 두 확장 계층의 양자화기를 단일 공유 양자화기로 만들어 QAT를 수행했다.

[0072] 본 발명의 실시예에 따른 로그레벨 임계값 양자화(LLTQ)은 학습된 단계 크기 양자화(LSQ) 방법[1]과 훈련된 양자화 임계값(TQT) 방법[2]은 균일한 양자화를 사용한다. 균일한 양자화기가 삽입된 CNN은 사전 훈련된 플로팅-포인트 파라미터를 사용하여 양자화 인식 훈련(QAT)을 수행한다.

[0073] 도 3은 균일한 양자화기가 삽입된 컨볼루션 계층의 구조와 양자화기의 4단계 작동을 보여준다. 첫 번째 단계인 스케일 프로세스, 양자화기는 실제 값 범위를 지정된 비트의 정수 값 범위에 매핑한다. 라운드 프로세스라고 하는 두 번째 단계는 정수 값 범위로 매핑된 모든 실수 값을 정수로 변환한다. 세 번째 단계는 클램프 프로세스이다. 이 프로세스는 양자화 수준에 맞게 양자화 범위를 초과하는 요소를 제거한다. 양자화 비트가 n인 경우, 클램프 프로세스 후의 값은 식(1)에 표시된 바와 같이 Bound 내부에 위치한다.

$$\text{Bound} = \begin{cases} [-2^{n-1}, 2^{n-1} - 1], & \text{signed data} \\ [0, 2^n - 1], & \text{unsigned data.} \end{cases} \quad (1)$$

[0074] 네 번째 단계인 de-quant 프로세스는 정수 범위에 매핑된 값을 양자화의 영향을 받는 플로팅-포인트 값으로 변환한다. 양자화된 값을 정수로 다시 플로팅-포인트로 변환하는 이유는 GPU를 사용하는 대부분의 기존 딥러닝 프레임워크가 플로팅-포인트 훈련에 최적화되어 있기 때문이다. 순방향 경로에 대한 전체 식은 다음과 같이 설명할 수 있다.

$$x_d = \text{clamp} \left(\text{round} \left(\frac{x_f}{s} \right), \text{Bound} \right) \cdot s \quad (2)$$

[0075] 여기서 x_f는 양자화기의 입력인 플로팅-포인트 값이고 x_d는 de-quant 프로세스를 통과한 양자화기의 출력 값이다. clamp(x, Bound) 함수는 균일 양자화기의 두 번째 단계이며 반올림된 값(x)을 경계 내에 맞도록 자른다. round(·) 함수는 균일 양자화기의 세 번째 단계이다. 실제 데이터를 가장 가까운 정수로 반올림한다.

straight-through 추정기(STE)[8]를 사용하여 반올림 공정에서 발생하는 불연속성을 해결하였다.

LSQ [1] 방법은 훈련 가능한 파라미터를 기준화 인수로 설정하고 다른 파라미터와 함께 기준화 인수를 훈련시킨다. 따라서 LSQ는 양자화 공정에서 기준화 인수를 변환할 필요가 없다. 섹션 III-B에 기술된 경량 CNN에 사용하기 위해 잘 알려져 있지만 하드웨어에 이를 구현할 때 전용 연산자가 필요하다는 단점이 있다. 이는 플로팅-포인트 기준화 인수이기 때문이다. TQT [2] 방법은 훈련 가능한 파라미터를 로그 임계값($\log_2 t$)으로 설정하고 기준화 인수를 다음과 같이 2의 거듭제곱 항에 매핑한다.

$$s = \frac{2^{\text{ceil}(\log_2 t)}}{q_level} \quad (3)$$

여기서 q_level 은 (1)의 Bound 최대값에 1을 더하는 값이다.

TQT는 기준화 인수를 2의 거듭제곱 항에 매핑하기 때문에 기준화 인수를 사용한 곱셈과 나눗셈을 이동 연산으로 대체할 수 있다는 장점이 있다. 단, 재훈련을 시작하기 전에 활성화 기준화 인수의 보정이 필요하다. 따라서 본 논문에서는 이전 연구 [1], [2]의 문제를 해결하는 새로운 하드웨어 친화적 양자화 방법을 제안한다. 제안된 LLTQ는 훈련 가능한 파라미터를 직접 2의 거듭제곱 인수에 매핑할 수 있다. 또한 TQT[2]와 달리 LLTQ 방법을 사용하면 첫 번째 epoch의 첫 번째 배치의 활성화와 데이터 통계 값으로 활성화 기준화 인수를 초기화할 수 있으므로 사전 보정 프로세스가 필요하지 않다. 또한 기준화 인수를 2의 거듭제곱 항에 매핑함으로써 스케일에서 발생하는 곱셈 및 나눗셈 연산과 de-quant 프로세스는 모두 이동 연산으로 대체될 수 있다. LLTQ 방법의 성능은 섹션 III-C-2에서 설명하는 정수 스케일 변환 방법을 통한 정수 산술만을 사용하여 평가된다. LLTQ는 훈련 가능한 파라미터를 레벨 임계값($level_th$)으로 설정하고 다음 식을 사용하여 기준화 인수를 2의 거듭제곱 항에 매핑한다.

$$s = \frac{2^{\text{ceil}(\log_2(\text{abs}(level_th * q_level))}}{q_level} \quad (4)$$

여기서 q_level 은 (3)에서와 같다. LLTQ 식과 식의 훈련 가능한 파라미터에 대한 자세한 설명은 보충 자료 문서의 부록 A에 제시되어 있다. 기준화 인수를 계산하려면 많은 연산이 필요하다. 그러나 정수 산술만 사용하도록 재구성된 CNN을 사용하여 추론을 수행할 경우 2의 거듭제곱 항에 매핑된 기준화 인수의 이동량만 필요하다.

또한 LLTQ에 대해 하드웨어 친화적인 양자화 방식이 여러 개 적용되었다. 대칭 양자화를 사용함으로써 균일한 양자화 중에 발생하는 영점에 대한 계산 오버헤드가 제거된다. 계층별 스케일링을 도입함으로써 주어진 파라미터 또는 입력 텐서의 모든 요소에 대해 하나의 기준화 인수로 양자화를 수행할 수 있다. 컨볼루션 층과 BN 계층의 파라미터를 결합한 BN 계층 융합이 적용되었다. 이 방식은 BN 계층에서 양자화를 진행할 필요가 없다. 이는 BN 계층에 필요한 곱셈과 나눗셈 연산을 제거함으로써 연산의 수와 추론 시간을 줄일 수 있다. fire 모듈은 스쿼즈 계층의 출력이 두 확장 계층의 입력으로 들어가는 구조이다. 플로팅-포인트에서는 두 확장 계층의 입력이 동일하다. 그러나 QAT에서 각 컨볼루션 계층에는 서로 다른 기준화 인수가 필요하다. 따라서 스쿼즈 계층은 각 확장 계층의 출력을 별도로 계산해야 한다. 이는 하드웨어에서 CNN 추론을 수행할 때 병목 현상을 일으킬 수 있다. 이 문제를 해결하기 위해 두 확장 계층의 양자화기를 단일 공유 양자화기로 만들어 QAT를 수행했다.

도 4는 본 발명의 일 실시예에 따른 정수 스케일 변환 방법을 설명하기 위한 도면이다.

제안하는 양자화 방법에 의해 만들어진 파라미터를 사용한 CNN은 제안하는 정수 스케일 변환 방법을 통해 정수 산술만을 사용하여 재구성되었다. 도 4(b) 및 도 4(d)와 같이 W_k^f , B_k^f , W_{k+1}^f , B_{k+1}^f , W_{k+2}^f , B_{k+2}^f 의 화살표는 플로팅-포인트 데이터의 흐름을 나타내고 나머지 화살표는 정수 데이터의 흐름을 나타낸다. 각 문자의 위첨자는 데이터 유형을 나타낸다. f 는 플로팅-포인트 데이터를 나타내고 i 는 정수를 나타내며 아래 첨자 k , $k+1$, $k+2$, $k+3$ 은 각 계층의 수를 나타낸다. Q_a 는 정수-산술 전용(Integer-Arithmetic-Only; IAO) 활성화 양자화기, Q_b 는 IAO 편향 양자화기, Q_w 는 IAO 가중치 양자화기이다. $S_{w,k}$, $S_{w,k+1}$, $S_{w,k+2}$ 는 가중치 기준화 인수, $S_{b,k}$, $S_{b,k+1}$, $S_{b,k+2}$ 는 편향 기준화 인수, $S_{a,k}$, $S_{a,k+1}$, $S_{a,k+2}$, $S_{a,k+3}$ 은 활성화 기준화 인수이다. 여기서 데이터 흐름 화살표에 표시된 A, W, B, C는 각각 계층 입력, 가중치, 편향, 활성화 함수 출력을 의미한다. 도 4에서 오프라인(Offline)으로 표시

된 부분은 양자화에 영향을 받는 플로팅-포인트 파라미터를 정수로 변환하는 과정이다. 이 프로세스는 추론 전에 수행되므로 추론 시간에 영향을 미치지 않는다. 양자화에 영향을 받는 플로팅-포인트 파라미터를 정수 파라미터로 변환하는 프로세스는 다음과 같이 표시된다.

$$W_k^i = \text{clamp}(\text{round}(W_k^f \ll S_{w,k}, \text{Bound})) \quad (5)$$

$$B_k^i = \text{clamp}(\text{round}(B_k^f \ll S_{b,k}, \text{Bound})) \\ \ll (S_{a,k} + S_{w,k} - S_{b,k}). \quad (6)$$

컨볼루션 계층의 입력은 QAT에서 지정한 비트 정밀도로 양자화된 정수라고 가정했다. 도 4(a)와 도 4(b)에서와 같이, 일반 컨볼루션 계층은 정수 스케일 가중치와 정수 스케일 활성화에서 MAC 연산을 수행하고 정수 스케일 편향을 MAC 결과에 추가한다. 식(6)에 나타난 것처럼 편향 값은 MAC 작동 결과와 동일한 스케일로 변환되어야 한다. 이러한 방법을 편향 업스케일링이라고 하며, 이 프로세스가 없으면 편향과 MAC 작동 결과가 비트 정밀도가 다르기 때문에 오류가 발생한다. 편향을 추가한 후 데이터는 활성화 함수 ReLU를 통과하며 ReLU의 결과는 식(7)과 같이 나타낼 수 있다.

$$C_k^i = \text{ReLU}(\text{MAC}(A_k^i, W_k^i) + B_k^i) \quad (7)$$

여기서 C_k^i 는 ReLU의 출력값이고, 현재 계층의 비트 정밀도에서 변환되어야 한다. MAC 연산은 다음 계층으로 입력되기 전에 QAT에서 지정한 비트 정밀도로 수행된다. 이 방법을 활성화 다운스케일링이라고 하며 현재 계층의 기준화 인수에 대한 영향을 취소하고 다음 계층의 입력 정밀도로 스케일링한다. 활성화 다운스케일링은 식(8)과 같이 나타낼 수 있다.

$$A_{k+1}^i = \text{clamp}(C_k^i \gg S_{a,k} + S_{w,k} - S_{a,k+1}, \text{Bound}) \quad (8)$$

도 4(c)와 도 4(d)에 표시된 fire 모듈에서 스쿼즈 계층의 습성은 식(5), 식(6), 식(7)이 적용되는 일반 컨볼루션 계층과 동일하다. 공유 양자화기의 도입으로 인해 단일 기준화 인수로 충분하다. 따라서 각 확장 계층에 대한 입력을 계산하지 않고도 동일한 입력을 사용할 수 있다. 각 확장 계층의 입력은 식(9)과 같이 설명될 수 있다.

$$A_{k+1}^i, A_{k+2}^i = \text{clamp}(C_k^i \gg S_{a,k} + S_{w,k} - S_{a,k+1}, \text{Bound}) \quad (9)$$

식(8)을 사용하여 활성화 다운스케일링 후 각 확장 계층의 출력을 채널별로 연결해야 한다. 채널별 연결은 값에 영향을 미치지 않으므로 확장이 필요하지 않다. 이 프로세스는 식(10)에 표시된다.

$$A_{k+3}^i = \text{concat}[\text{clamp}(C_{k+1}^i \gg S_{a,k+1} + S_{w,k+1} - S_{a,k+3}, \text{Bound}), \text{clamp}(C_{k+2}^i \gg S_{a,k+2} + S_{w,k+2} - S_{a,k+3}, \text{Bound})]. \quad (10)$$

도 5는 본 발명의 일 실시예에 따른 실시간 얼굴 감정 인식을 위한 자원 효율적인 정수-산술 전용 FPGA 기반 CNN 가속기의 아키텍처를 나타낸다.

본 발명의 실시예에 따른 실시간 얼굴 표정 인식을 위한 CNN 가속기는 CPU(Processing System; PS)(510)와 FPGA(Programmable Logic; PL)(520)로 구성된 이기종 SoC 플랫폼에서 구현되었다. CPU(510)는 메모리 영역을 할당하고 각 계층 구성 정보인 커널 크기, 스트라이드, 패딩, 입력 채널, 출력 채널, 이동량 및 주소를 AXI_LITE 버스를 통해 FPGA(520) 가속기 IP로 전송한다. 입력 이미지, 특징맵, 파라미터와 같이 MAC 연산을 위한 데이터는 AXI_MEM 버스를 통해 DRAM에서 FPGA(520) 가속기로 전송된다. CNN 공유를 구성하는 각 계층은 작동을 수행하기 위해 다중 누적 단위(MACU)와 후처리 모듈(PPM)을 공유한다. 연산 결과는 AXI_MEM 버스를 사용하여 FPGA(520) 가속기에서 DRAM으로 전달된다.

본 발명의 실시예에 따른 범용 임베디드 CPU는 6개의 중첩된 루프가 있는 컨볼루션 작업을 수행하는 속도가 매

우 느리다. 이 문제를 해결하기 위해 SoC 플랫폼의 PL 영역(520)에 병렬 처리 모듈을 구현하여 CNN 추론 프로세스를 가속화했다. 병렬 처리 모듈을 구현하기 위해 병렬 처리 전략을 6개의 중첩 루프에 적용할 수 있는 부분을 찾아 한다. 여러 가지 가능한 병렬화 전략 중 두 가지가 임베디드 장치에서 루프 수준 병렬화를 효율적으로 구현하기 위해 사용되었다[9].

[0103] 첫째, 컨볼루션 내 병렬화 전략이 실행되었다. 도 5에 도시된 MACU 내부 작동에 해당하는 루프를 완전 언롤링(unrolling)함으로써 컨볼루션 내 병렬화를 달성하였다. 완전 언롤링된 루프(fully unrolled loop)는 9개의 병렬 승수와 3×3 컨볼루션 연산을 위한 8개의 가산기로 구성된 MACU로 구현되었다. 둘째, 특징맵 병렬화 전략이 적용되었다. 도 5에 도시된 MACU는 출력 특징맵의 다른 채널에 대한 루프를 부분적으로 해제하여 특징맵 병렬화를 달성하기 위해 사용되었다. 부분적 풀린 루프의 언롤링(unrolling) 인수가 4였기 때문에, 가속기는 출력 특징맵의 다른 4개 채널에 대해 MAC 연산을 병렬로 실행할 수 있다. 1×1 연산은 3×3 컨볼루션 연산을 위해 완전 언롤링된(fully unrolled) MACU의 일부를 재사용함으로써 수행된다.

[0104] SoC 플랫폼을 고려할 때, 더 많은 병렬화 전략을 사용할 수 있다. 그러나, 특징맵 병렬 처리의 경우, 동일한 출력 채널의 픽셀을 대상으로 하는 각 입력 채널에 대한 MAC 연산의 모든 결과를 축적해야 했기 때문에 데이터 의존성으로부터 자유롭지 못했다. 따라서 많은 리소스가 활용될 가능성이 높다. 컨볼루션 간 병렬화의 경우, 많은 양의 데이터가 온칩 버퍼에 미리 로드되어야 하기 때문에 임베디드 장치에서의 구현에는 적합하지 않았다.

[0105] 본 발명의 실시예에 따른 데이터 캐싱 모듈에 있어서, MAC 연산을 위해 매번 오프칩 메모리에서 데이터에 액세스하는 프로세스는 대기 시간과 에너지 측면에서 비용이 많이 든다. 또한 PL 영역(520) 내부의 온칩 메모리는 CNN의 모든 파라미터와 계산 결과를 저장할 만큼 크지 않다. 따라서 온칩 메모리에서 데이터 재사용을 극대화하여 오프칩 메모리에 대한 액세스 수를 최소화할 필요가 있다. PL 영역(520)에는 3개의 데이터 캐싱 모듈(FMCM, PCM, OCAM)이 사용되었다.

[0106] 컨볼루션 내 병렬 처리 전략이 적용된 MACU에서 3×3 컨볼루션 필터는 MAC 작동을 위해 9픽셀의 입력 특징맵을 필요로 한다. 입력 특징맵의 효율적인 미리 읽어오기를 위해 가속기에는 적어도 3개의 라인을 저장할 수 있는 라인 버퍼가 필요하다. 주소 지정에 편리하도록 입력 특징맵의 4개 라인을 저장할 수 있는 라인 버퍼가 만들어졌다. 이러한 라인 버퍼는 함께 번들되었으며 특징맵 캐싱 모듈(FMCM)이라고 불렸다. 파라미터 데이터 재사용을 극대화하기 위해, 파라미터 캐싱 모듈(PCM)이라 불리는, 작업이 수행되는 계층의 모든 파라미터를 저장할 수 있는 온칩 메모리가 생성되었으며, 각 프로세싱 모듈의 PCM을 분할하여 특징맵 병렬화 전략을 사용하였다. 메모리는 MAC 연산 중에 출력 특징맵의 채널에 대한 MAC 연산 결과를 축적하기 위해 생성되었으며, 메모리는 출력 채널 축적 모듈(OCAM)이라고 한다.

[0107] 본 발명의 실시예에 따른 후처리 모듈에 있어서, 컨볼루션 연산을 제외한 나머지 모든 연산은 PPM에서 수행된다. PPM은 이동 연산, 추가 연산 및 부호 비트 비교 연산으로 구성된다. PPM은 OCAM의 최종 출력 값을 입력으로 수신한다. 또한 병렬 처리 모듈처럼, 특징맵 병렬화 전략은 언롤링된(unrolled) 인수를 4로 설정한 상태에서 적용된다. PPM의 최종 출력은 다음 계층의 입력을 입력하기 위한 DRAM으로의 후기입(writeback)이다.

[0109] 도 6은 본 발명의 일 실시예에 따른 정수-산술 전용 FPGA 기반 CNN 가속기에 대한 전체 작업 스케줄링 과정을 설명하기 위한 도면이다.

[0110] 도 6을 참조하면 첫째, CPU는 CNN 작업을 시작하기 전에 미리 계산된 네트워크 제약 조건을 사용하여 데이터 캐싱 모듈의 크기와 계층 및 데이터 영역의 주소를 계산한다(610). 계산된 주소와 계층 구성은 AXI_LITE 버스를 통해 PS에서 PL 구성 레지스터로 전송된다. 데이터 캐싱 모듈(FMCM, PCM)은 사전 전송된 주소 및 계층 구성을 사용하여 AXI_MEM 버스를 통해 DRAM에서 데이터를 사전 추출한다(620). 미리 읽기된 데이터는 픽셀 버퍼와 파라미터 버퍼에 로드되고 MAC 연산은 병렬 처리 모듈에서 수행된다(630). OCAM은 MAC 연산의 중간 결과를 저장하고 축적한다. 축적 완료 후 PPM은 나머지 연산을 실행한다(640). 이 데이터 흐름은 계층의 모든 프로세스가 종료될 때까지 지속된다(650).

[0111] 앞서 설명된 바와 같이, 본 발명에서는 실시간 얼굴 표정 인식을 위한 FPGA 기반 CNN 가속기를 제안한다. 가속기는 임베디드 시스템에 최적화되었으며 새로운 하드웨어 친화적인 양자화 방법을 적용하였다.

[0112] 본 발명의 실시예에 따르면, 얼굴 표정 인식 성능을 향상시키기 위해 FERPlus-A를 제시한다. 이것은 양방향 필터링, CLAHE(Contrast Limited Adaptive History Equalization) 및 엣지 강화와 같은 다양한 이미지 처리 알고

리즘을 적용하여 생성된 새로운 훈련 데이터셋이다.

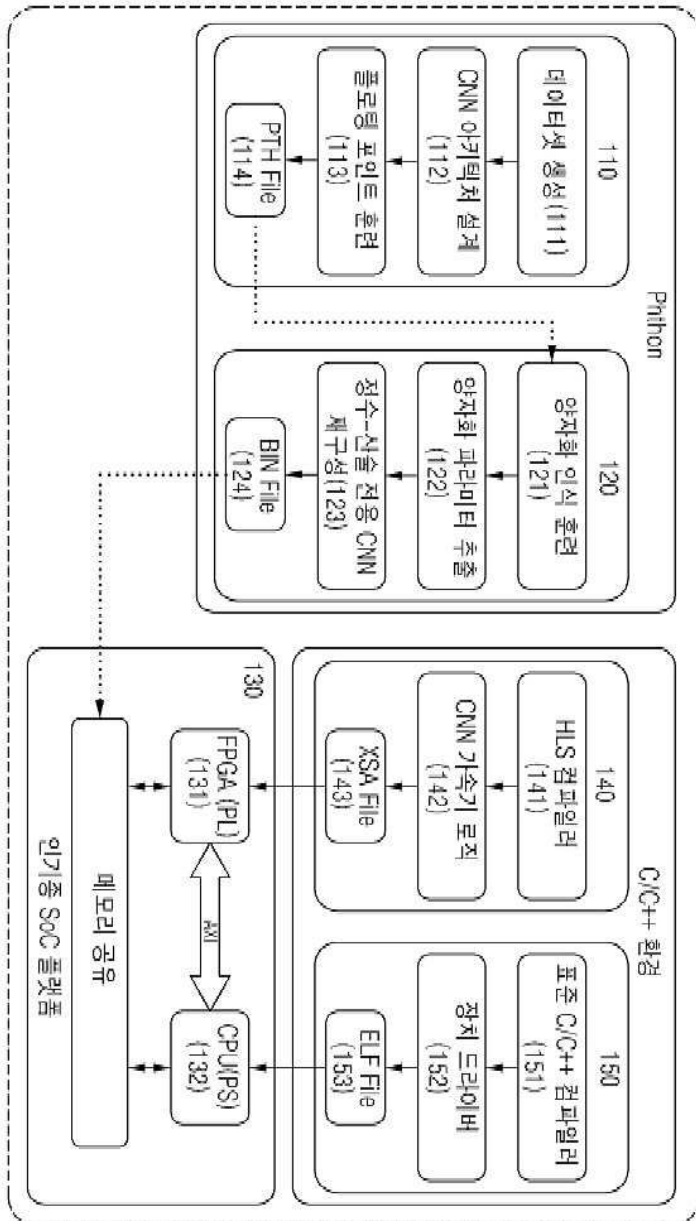
- [0113] 본 발명의 실시예에 따르면, 임베디드 장치를 위해 경량 CNN 아키텍처를 제안한다. 제안된 CNN은 스퀴즈넷(SqueezeNet)의 수정된 파이어(fire) 모듈을 기본 계산 모듈로 사용하여 작동 수와 메모리 설치 공간을 최적화한다. 제안된 CNN은 FERPlus 테스트 데이터셋을 사용하여 평가할 때 약 86.58%의 얼굴 표정 인식 정확도를 보였다. 이는 경량 CNN을 사용한 이전 작업과 비교하여 보고된 가장 높은 성능이다.
- [0114] 본 발명의 실시예에 따르면, 로그 레벨 임계값 양자화(Log Level Threshold Quantization; LLTQ) 방법을 제안하고, 이는 새로운 하드웨어 친화적인 양자화 방법이다. 제안된 방법은 기준화 인수를 사용한 곱셈 및 나눗셈 연산을 이동 연산으로 대체할 수 있다. 제안된 양자화 CNN 파라미터의 크기는 약 0.39MB이며, 작동 횟수는 약 28M 정수 연산(Integer Operations; IOP)이다. 또한 8비트 정수 산술만을 사용하는 CNN의 성능은 32비트 플로팅-포인트(floating-point) 사용에 비해 정확도가 떨어지지 않는 것을 확인하였다.
- [0115] 본 발명의 실시예에 따르면, 임베디드 장치에 최적화된 제안된 CNN은 실시간 얼굴 표정 인식을 위한 병렬화 전략과 효율적인 데이터 캐싱 전략을 적용하여 Xilinx ZC706 평가판에서 구현된다. FPGA 기반 CNN 가속기, 8비트 정수 산술만 사용하며, LUT 5,090개, 플립플롭 7,588개, BRAM 61개, DSP 49개를 활용했다. 가속기는 2.3W의 전력을 소모하면서 대략 초당 10 프레임(Frame per Second; FPS)의 프레임률을 달성했다.
- [0117] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 컨트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPA(field programmable array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.
- [0118] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embodiment)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.
- [0119] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.
- [0120] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형

태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

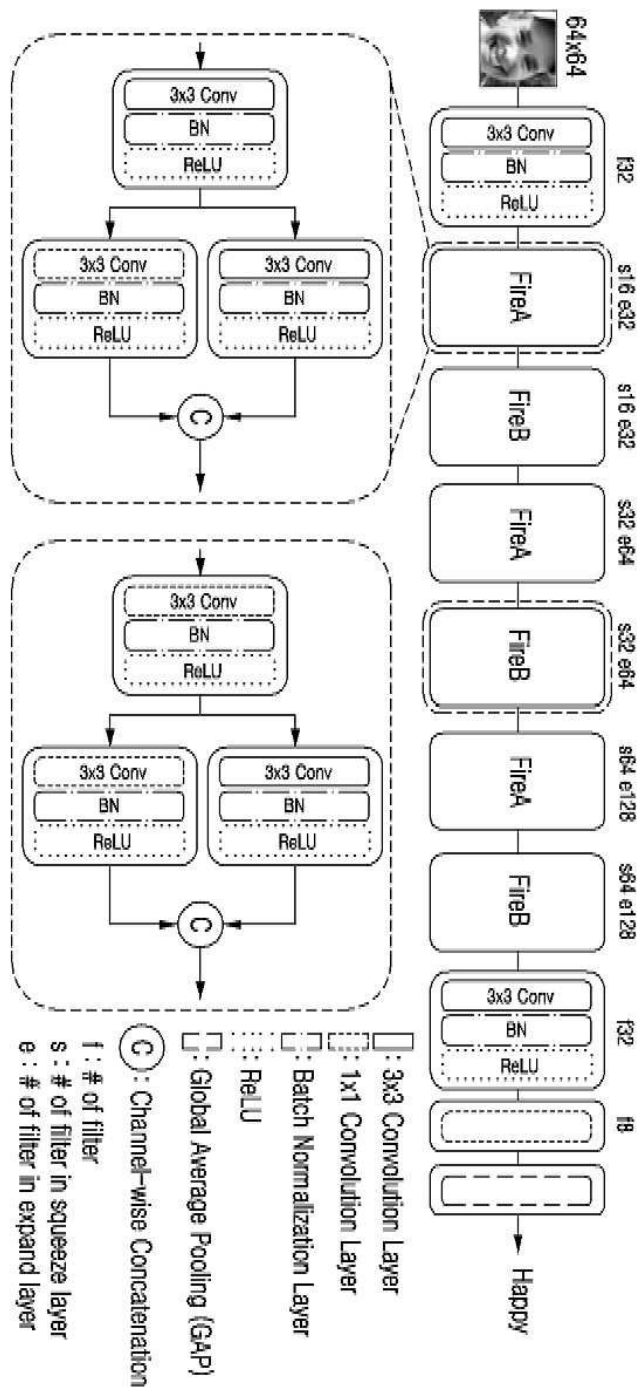
- [0121] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.
- [0123] <참고문헌>
- [0124] [1] S. K. Esser, J. L. McKinstry, D. Bablani, R. Appuswamy, and D. S. Modha, "Learned step size quantization," 2019, arXiv:1902.08153. [Online]. Available: <https://arxiv.org/abs/1902.08153>
- [0125] [2] S. R. Jain, A. Gural, M. Wu, and C. H. Dick, "Trained quantization thresholds for accurate and efficient fixed-point inference of deep neural networks," in Proc. Mach. Learn. Syst., Mar. 2020, pp. 112-128.
- [0126] [3] H. Phan-Xuan, T. Le-Tien, and S. Nguyen-Tan, "FPGA platform applied for facial expression recognition system using convolutional neural networks," Procedia Comput. Sci., vol. 151, pp. 651-658, Jan. 2019.
- [0127] [4] T. N. D. Phuc, N. N. Nhut, N. Trinh, and L. Tran, "Facial expression recognition system using FPGA-based convolutional neural network," in Research in Intelligent and Computing in Engineering, vol. 1254. Singapore: Springer, Jan. 2021, pp. 341-351.
- [0128] [5] P. T. Vinh and T. Q. Vinh, "Facial expression recognition system on SoC FPGA," in Proc. Int. Symp. Electr. Electron. Eng. (ISEE), Ho Chi Minh City, Vietnam, Oct. 2019, pp. 1-4.
- [0129] [6] R. Ding, G. Su, G. Bai, W. Xu, N. Su, and X. Wu, "A FPGA-based accelerator of convolutional neural network for face feature extraction," in Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits (EDSSC), Xi'an, China, Jun. 2019, pp. 1-3.
- [0130] [7] R. Ding, X. Tian, G. Bai, G. Su, and X. Wu, "Hardware implementation of convolutional neural network for face feature extraction," in Proc. IEEE 13th Int. Conf. ASIC (ASICON), Chongqing, China, Oct. 2019, pp. 1-4.
- [0131] [8] Y. Bengio, N. Leonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013, arXiv:1308.3432. [Online]. Available: <https://arxiv.org/abs/1308.3432>
- [0132] [9] K. Abdelouahab, M. Pelcat, J. Serot, and F. Berry, "Accelerating CNN inference on FPGAs: A survey," 2018, arXiv:1806.01683. [Online]. Available: <https://arxiv.org/abs/1806.01683>
- [0133] [10] L. Zhu. THOP: PyTorch-OpCounter. Accessed: Apr. 26, 2021. [Online]. Available: <https://github.com/Lyken17/pytorch-OpCounter>

도면

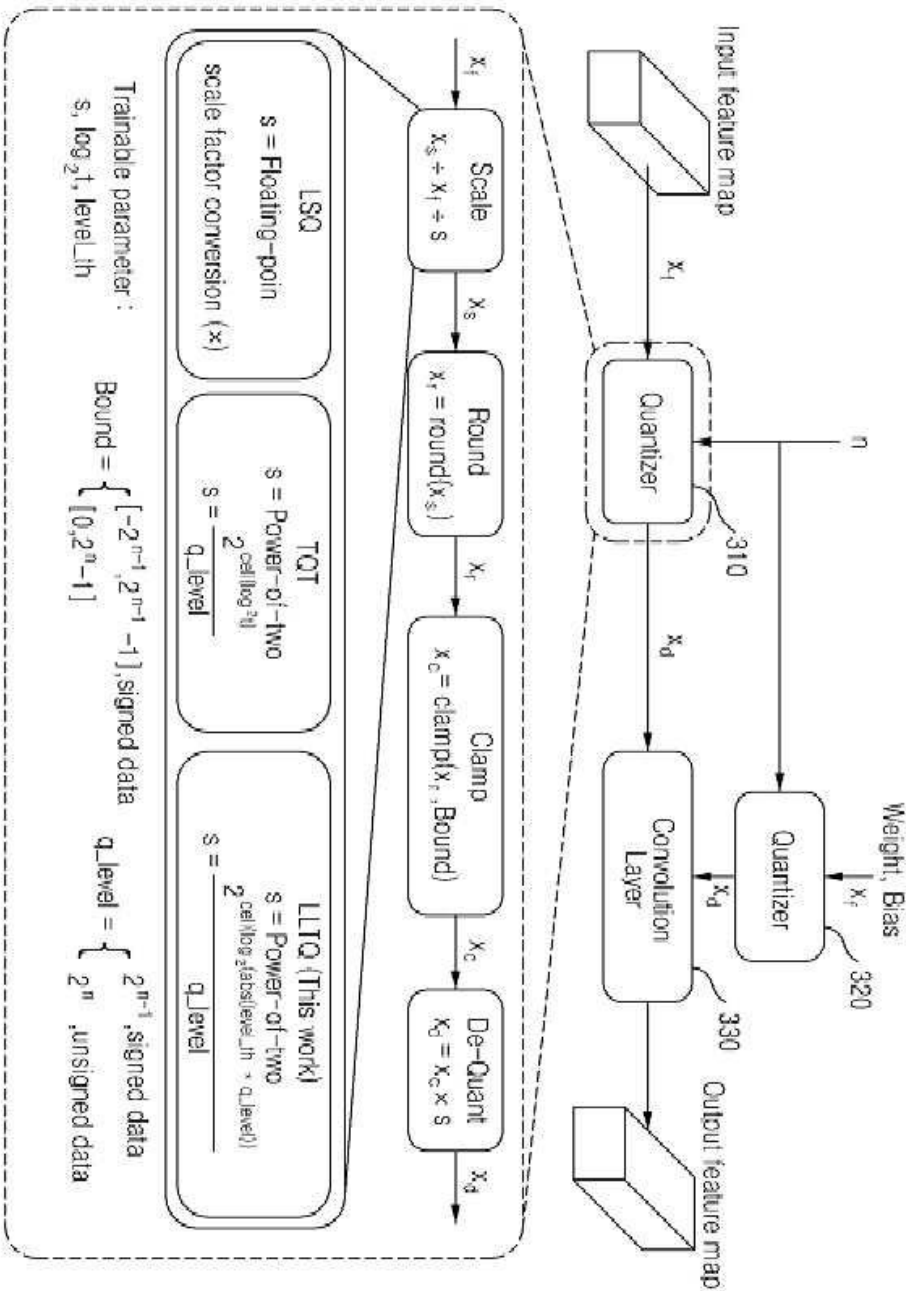
도면1



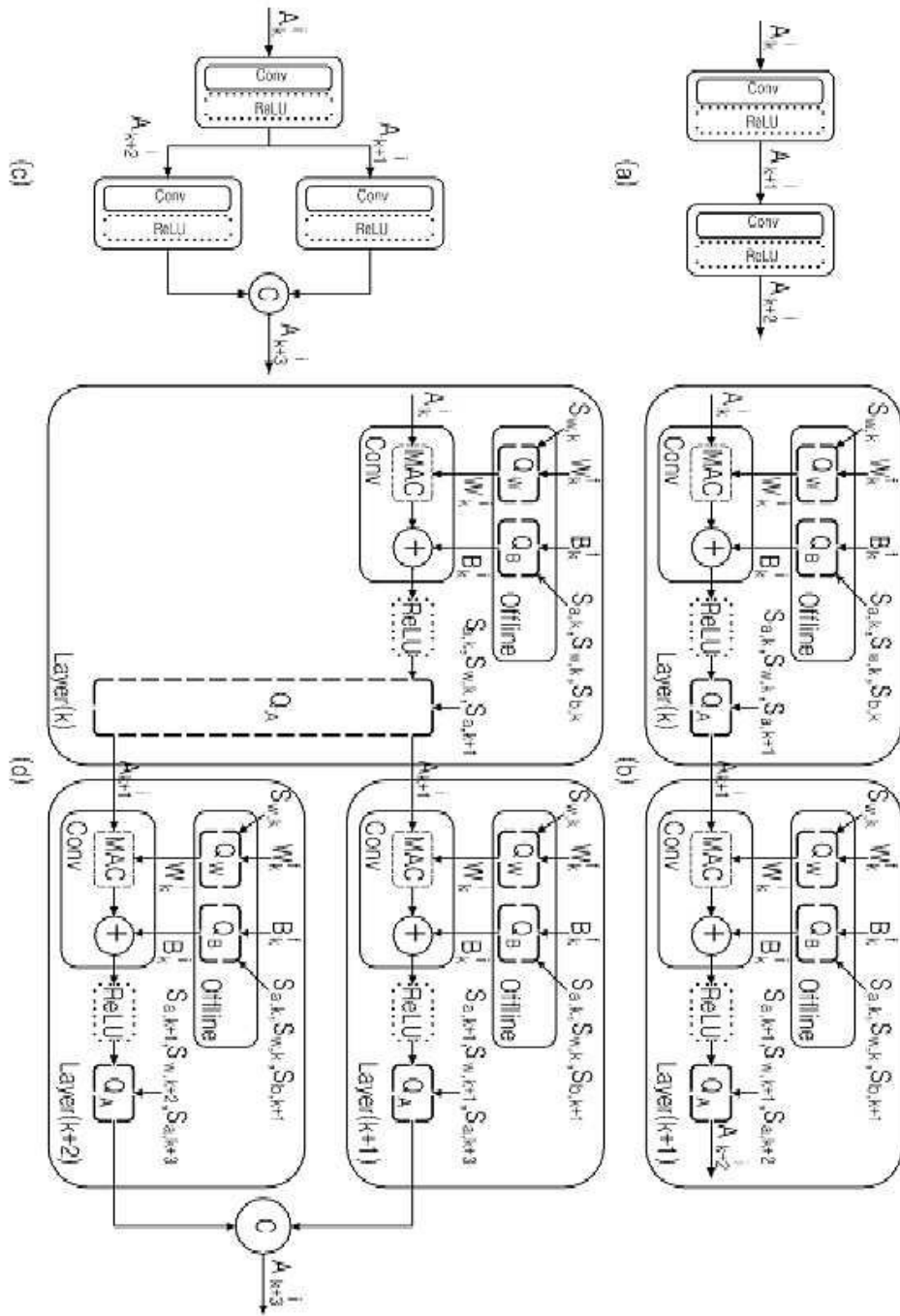
도면2



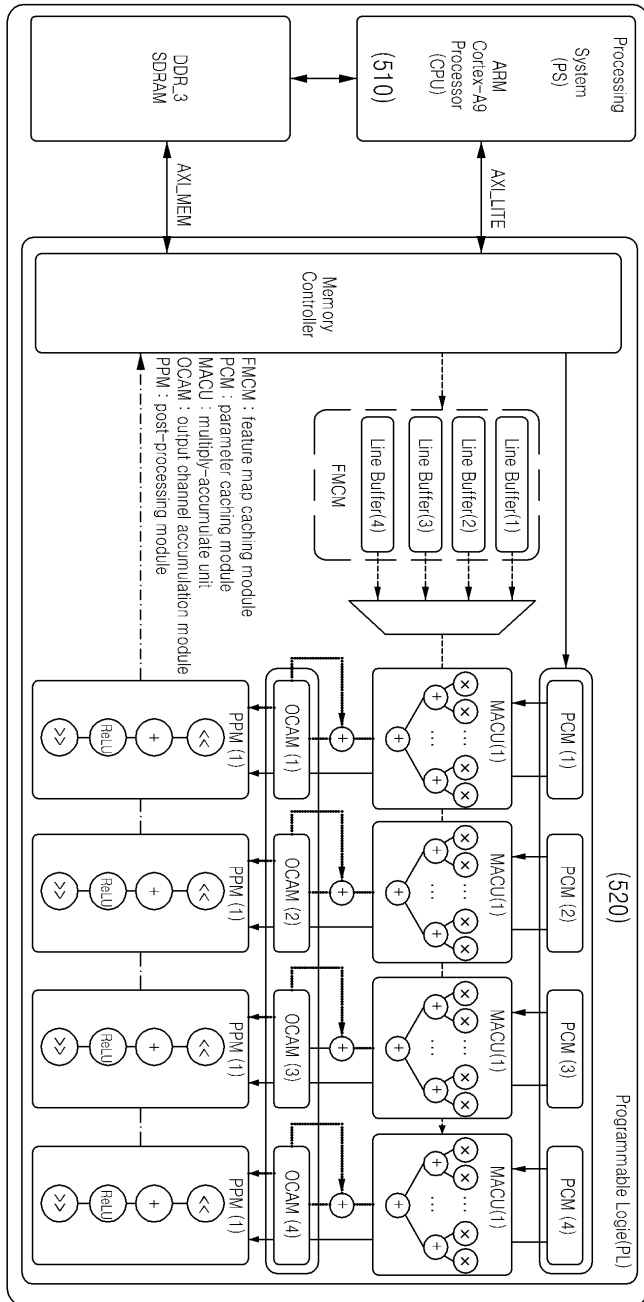
도면3



도면4



도면5



도면6

