



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0090145
(43) 공개일자 2024년06월21일

(51) 국제특허분류(Int. Cl.)

H04N 19/60 (2014.01) G06N 3/0455 (2023.01)
G06N 3/096 (2023.01) G06N 3/0985 (2023.01)
H04N 19/147 (2014.01) H04N 19/19 (2014.01)
H04N 19/44 (2014.01) H04N 19/91 (2014.01)

(52) CPC특허분류

H04N 19/60 (2015.01)
G06N 3/0455 (2023.01)

(21) 출원번호 10-2024-7009071

(22) 출원일자(국제) 2022년09월15일

심사청구일자 없음

(85) 번역문제출일자 2024년03월18일

(86) 국제출원번호 PCT/US2022/076496

(87) 국제공개번호 WO 2023/049655

국제공개일자 2023년03월30일

(30) 우선권주장

17/486,732 2021년09월27일 미국(US)

(71) 출원인

퀄컴 인코포레이티드

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

(72) 발명자

주 인하오

미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775

양 양

미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775

코현 타코 세바스티안

미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(74) 대리인

특허법인코리아나

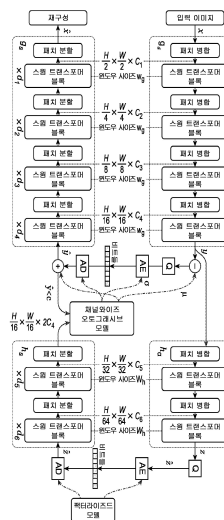
전체 청구항 수 : 총 49 항

(54) 발명의 명칭 미디어의 변환 코딩을 위한 트랜스포머 기반 아키텍처

(57) 요약

신경망 시스템을 사용하여 미디어 데이터를 프로세싱하기 위한 시스템 및 기법이 본 명세서에서 설명된다. 예를 들어, 프로세스는, 인코딩된 이미지 데이터의 프레임의 잠재 표현을 획득하는 것 및 입력으로서 인코딩된 이미지 데이터의 프레임의 잠재 표현을 사용하여 디코더 서브네트워크의 복수의 디코더 트랜스포머 레이어들에 의해, 디코딩된 이미지 데이터의 프레임을 생성하는 것을 포함할 수 있다. 복수의 디코더 트랜스포머 레이어들 중 적어도 하나의 디코더 트랜스포머 레이어는 특징들의 하나 이상의 패치들을 생성하고 하나 이상의 패치들 상에 적용된 하나 이상의 윈도우 파티션들 및 시프트된 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하기 위한 하나 이상의 트랜스포머 블록들; 및 하나 이상의 패치들 중 각각의 패치의 개개의 사이즈를 줄이기 위한 패치 병합 해제 엔진을 포함한다.

대표도



(52) CPC특허분류

G06N 3/096 (2023.01)

G06N 3/0985 (2023.01)

H04N 19/147 (2015.01)

H04N 19/19 (2015.01)

H04N 19/44 (2015.01)

H04N 19/91 (2015.01)

명세서

청구범위

청구항 1

미디어 데이터를 프로세싱하기 위한 장치로서,

적어도 하나의 메모리; 및

상기 적어도 하나의 메모리에 커플링된 적어도 하나의 프로세서를 포함하고, 상기 적어도 하나의 프로세서는:

인코딩된 이미지 데이터의 프레임의 잠재 표현을 획득하고;

입력으로서 상기 인코딩된 이미지 데이터의 프레임의 잠재 표현을 사용하여 디코더 서브네트워크의 복수의 디코더 트랜스포머 레이어들에 기초하여, 디코딩된 이미지 데이터의 프레임을 생성하도록 구성되고, 상기 복수의 디코더 트랜스포머 레이어들 중 적어도 하나의 디코더 트랜스포머 레이어는

특징들의 하나 이상의 패치들을 생성하고 상기 하나 이상의 패치들 상에 적용된 하나 이상의 윈도우 파티션들 및 시프트된 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하도록 구성된 하나 이상의 트랜스포머 블록들; 및

상기 하나 이상의 패치들 중 각각의 패치의 개개의 사이즈를 줄이도록 구성되는 패치 병합 해제 엔진

을 포함하는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 2

제 1 항에 있어서,

상기 디코딩된 이미지 데이터의 프레임을 생성하기 위해, 상기 적어도 하나의 프로세서는

상기 복수의 디코더 트랜스포머 레이어들 중 제 1 디코더 트랜스포머 레이어의 제 1 트랜스포머 블록을 사용하여, 상기 하나 이상의 패치들 상에 적용된 하나 이상의 제 1 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하고;

상기 제 1 디코더 트랜스포머 레이어의 제 2 트랜스포머 블록을 사용하여, 상기 하나 이상의 패치들 상에 적용된 하나 이상의 제 2 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것으로서, 상기 하나 이상의 제 2 윈도우 파티션들은 상기 하나 이상의 제 1 윈도우 파티션들 중 인접한 것들 사이에 하나 이상의 경계들과 중첩되도록 시프트되는, 상기 셀프 어텐션을 결정하고;

상기 패치 병합 해제 엔진을 사용하여, 복수의 병합 해제된 패치들로 상기 하나 이상의 패치들의 각각의 패치를 세그먼트화하는 것으로서, 상기 복수의 병합 해제된 패치들은 중첩되지 않는, 상기 하나 이상의 패치들의 각각의 패치를 세그먼트화하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 3

제 2 항에 있어서,

상기 적어도 하나의 프로세서는 상기 복수의 디코더 트랜스포머 레이어들 중 제 2 디코더 트랜스포머 레이어의 제 1 트랜스포머 블록에 상기 복수의 병합 해제된 패치들을 제공하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 4

제 3 항에 있어서,

상기 적어도 하나의 프로세서는

상기 제 2 디코더 트랜스포머 레이어의 패치 병합 해제 엔진을 사용하여, 상기 복수의 병합 해제된 패치들을 세그먼트화하고;

상기 패치 병합 해제 엔진의 출력을 상기 복수의 디코더 트랜스포머 레이어들 중 제 3 디코더 트랜스포머 레이어에 제공하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 5

제 2 항에 있어서,

상기 복수의 병합 해제된 패치들 중 각각의 병합 해제된 패치는 균일한 패치 사이즈를 갖고 상기 패치 병합 해제 엔진은 2의 패치 사이즈 감소 팩터를 적용하는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 6

제 2 항에 있어서,

상기 복수의 병합 해제된 패치들로 상기 하나 이상의 패치들 중 각각의 패치를 세그먼트화하기 위해, 상기 적어도 하나의 프로세서는 상기 복수의 병합 해제된 패치들의 특징 차원(feature dimension)을 감소시키도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 7

제 1 항에 있어서,

상기 복수의 디코더 트랜스포머 레이어들은 상기 인코딩된 이미지 데이터의 프레임의 잠재 표현을 입력으로 수신하고 비선형 변환을 적용하여 디코딩된 이미지 데이터의 프레임을 생성하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 8

제 7 항에 있어서,

상기 비선형 변환은 합성 변환이고, 상기 디코딩된 이미지 데이터의 프레임은 상기 인코딩된 이미지 데이터의 프레임과 연관된 입력 이미지의 재구성인, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 9

제 1 항에 있어서,

상기 적어도 하나의 프로세서는 레이트-왜곡(rate-distortion)에 적어도 부분적으로 기초한 손실 함수를 사용하여 상기 복수의 디코더 트랜스포머 레이어들 중 하나 이상의 디코더 트랜스포머 레이어들을 훈련하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 10

제 9 항에 있어서,

상기 손실 함수는 레이트 왜곡을 위한 라그랑지 승수(Lagrangian multiplier)를 포함하는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 11

제 1 항에 있어서,

상기 적어도 하나의 디코더 트랜스포머 레이어에 포함된 상기 하나 이상의 트랜스포머 블록들 중 적어도 일부는 동일한 아키텍처를 갖는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 12

제 1 항에 있어서,

상기 적어도 하나의 디코더 트랜스포머 레이어에 포함된 상기 하나 이상의 트랜스포머 블록들의 각각은 동일한 아키텍처를 갖는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 13

제 1 항에 있어서,

상기 인코딩된 이미지 데이터의 프레임은 인코딩된 정지 이미지를 포함하는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 14

제 1 항에 있어서,

상기 인코딩된 이미지 데이터의 프레임은 인코딩된 비디오 프레임을 포함하는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 15

제 1 항에 있어서,

상기 적어도 하나의 프로세서는 적어도 제 1 훈련 데이터 세트 및 제 2 훈련 데이터 세트로 상기 복수의 디코더 트랜스포머 레이어들을 훈련하도록 구성되고, 상기 제 2 훈련 데이터 세트의 데이터는 상기 제 1 훈련 데이터 세트의 데이터와 비교하여 역 시간 순서(reversed temporal order)를 갖는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 16

제 1 항에 있어서,

상기 복수의 디코더 트랜스포머 레이어들은 일련의 연속적인 디코더 트랜스포머 레이어들을 포함하는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 17

미디어 데이터를 프로세싱하기 위한 장치로서,

적어도 하나의 메모리; 및

상기 적어도 하나의 메모리에 커플링된 적어도 하나의 프로세서를 포함하고, 상기 적어도 하나의 프로세서는: 프레임들 복수의 패치들로 세그먼트화하고;

입력으로서 상기 복수의 패치들을 사용하여 인코더 서브네트워크의 복수의 인코더 트랜스포머 레이어들에 기초하여, 인코딩된 이미지 데이터의 프레임을 생성하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 18

제 17 항에 있어서,

상기 인코딩된 이미지 데이터의 프레임을 생성하기 위해, 상기 적어도 하나의 프로세서는

상기 복수의 인코더 트랜스포머 레이어들 중 제 1 인코더 트랜스포머 레이어의 제 1 트랜스포머 블록을 사용하여, 하나 이상의 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하고;

상기 제 1 인코더 트랜스포머 레이어의 제 2 트랜스포머 블록을 사용하여, 하나 이상의 시프트된 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것으로서, 상기 하나 이상의 시프트된 윈도우 파티션들은 상기 하나 이상의 윈도우 파티션들과 중첩하는, 상기 셀프 어텐션을 결정하고;

상기 제 1 트랜스포머 블록 및 상기 제 2 트랜스포머 블록 중 하나 이상을 사용하여, 상기 세그먼트화된 프레임에 비선형 변환을 적용하기 위한 특징들의 하나 이상의 패치들을 결정하고;

패치 병합 엔진을 사용하여, 상기 제 1 인코더 트랜스포머 레이어와 제 2 인코더 트랜스포머 레이어 사이의 패치 사이즈를 증가시키도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 19

제 18 항에 있어서,

상기 패치 병합 엔진은 상기 제 1 인코더 트랜스포머 레이어로부터의 복수의 인접한 패치들을 상기 제 2 인코더 트랜스포머 레이어에 제공되는 병합된 패치로 결합하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 20

제 18 항에 있어서,

상기 제 1 인코더 트랜스포머 레이어의 상기 제 2 트랜스포머 블록의 출력은 상기 제 2 인코더 트랜스포머 레이어의 입력에 커플링되는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 21

제 18 항에 있어서,

상기 적어도 하나의 프로세서는

입력으로서 상기 복수의 패치들을 사용하여 상기 인코더 서브네트워크의 상기 복수의 인코더 트랜스포머 레이어들을 사용하여, 상기 세그먼트화된 프레임에 대한 계층적 특징 맵을 생성하고;

상기 계층적 특징 맵으로부터 상기 인코딩된 이미지 데이터의 프레임을 생성하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 22

제 18 항에 있어서,

상기 복수의 패치들 중 각각의 패치는 균일한 사이즈를 가지며 상기 세그먼트화된 프레임의 하나 이상의 픽셀들을 포함하는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 23

제 18 항에 있어서,

상기 패치 병합 엔진은, 인접한 패치들의 하나 이상의 서브세트들에서 얻은 특징들을 연쇄(concatenating)시켜 상기 패치 사이즈를 늘리도록 구성되고, 상기 인접한 패치들의 각각의 서브세트는 상기 패치 병합 엔진에 의해 병합된 패치 출력으로 병합되는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 24

제 18 항에 있어서,

상기 제 1 트랜스포머 블록 및 상기 제 2 트랜스포머 블록은 동일한 아키텍처를 갖는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 25

제 18 항에 있어서,

상기 적어도 하나의 프로세서는 상기 제 1 인코더 트랜스포머 레이어 이전에 상기 인코더 서브네트워크의 선형 임베딩 레이어에 상기 복수의 패치들을 제공하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 26

제 17 항에 있어서,

상기 인코딩된 이미지 데이터의 프레임은 이미지 데이터의 잠재 표현인, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 27

제 26 항에 있어서,

상기 잠재 표현은 상기 인코더 서브네트워크의 상기 복수의 인코더 트랜스포머 레이어들에 의해 생성된 계층적 특징 맵인, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 28

제 17 항에 있어서,

상기 적어도 하나의 프로세서는 레이트-왜곡 손실에 기초한 손실 함수를 사용하여 상기 복수의 인코더 트랜스포머 레이어들 중 하나 이상의 인코더 트랜스포머 레이어들을 훈련하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 29

제 28 항에 있어서,

상기 손실 함수는 레이트 왜곡을 위한 라그랑지 승수(Lagrangian multiplier)를 포함하는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 30

제 17 항에 있어서,

상기 복수의 패치들은 정지 이미지 프레임 또는 비디오 프레임을 포함하는 입력으로부터 세그먼트화되는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 31

제 17 항에 있어서,

상기 적어도 하나의 프로세서는 팩터라이즈드 프라이어를 이용하여 상기 인코딩된 이미지 데이터를 엔트로피 코딩하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.

청구항 32

미디어 데이터를 프로세싱하는 방법으로서,

인코딩된 이미지 데이터의 프레임의 잠재 표현을 획득하는 단계; 및

입력으로서 상기 인코딩된 이미지 데이터의 프레임의 잠재 표현을 사용하여 디코더 서브네트워크의 복수의 디코더 트랜스포머 레이어들에 의해, 디코딩된 이미지 데이터의 프레임을 생성하는 단계를 포함하고, 상기 복수의 디코더 트랜스포머 레이어들 중 적어도 하나의 디코더 트랜스포머 레이어는

특징들의 하나 이상의 패치들을 생성하고 상기 하나 이상의 패치들 상에 적용된 하나 이상의 윈도우 파티션들 및 시프트된 윈도우 파티션들 내에서 국부적으로 셀프 어텐션(self-attention)을 결정하기 위한 하나 이상의 트랜스포머 블록들; 및

상기 하나 이상의 패치들 중 각각의 패치의 개개의 사이즈를 줄이기 위한 패치 병합 해제 엔진

을 포함하는, 미디어 데이터를 프로세싱하는 방법.

청구항 33

제 32 항에 있어서,

상기 복수의 디코더 트랜스포머 레이어들에 의해, 상기 디코딩된 이미지 데이터의 프레임을 생성하는 단계는

상기 복수의 디코더 트랜스포머 레이어들 중 제 1 디코더 트랜스포머 레이어의 제 1 트랜스포머 블록에 의해, 상기 하나 이상의 패치들 상에 적용된 하나 이상의 제 1 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것;

상기 제 1 디코더 트랜스포머 레이어의 제 2 트랜스포머 블록에 의해, 상기 하나 이상의 패치들 상에 적용된 하나 이상의 제 2 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것으로서, 상기 하나 이상의 제 2 윈도우 파티션들은 상기 하나 이상의 제 1 윈도우 파티션들 중 인접한 것들 사이에 하나 이상의 경계들과 중첩

되도록 시프트되는, 상기 셀프 어텐션을 결정하는 것; 및

상기 패치 병합 해제 엔진에 의해, 복수의 병합 해제된 패치들로 상기 하나 이상의 패치들의 각각의 패치를 세그먼트화하는 것으로서, 상기 복수의 병합 해제된 패치들은 중첩되지 않는, 상기 하나 이상의 패치들의 각각의 패치를 세그먼트화하는 것을 포함하는, 미디어 데이터를 프로세싱하는 방법.

청구항 34

제 33 항에 있어서,

상기 복수의 디코더 트랜스포머 레이어들 중 제 2 디코더 트랜스포머 레이어의 제 1 트랜스포머 블록에 상기 복수의 병합 해제된 패치들을 제공하는 단계를 더 포함하는, 미디어 데이터를 프로세싱하는 방법.

청구항 35

제 34 항에 있어서,

상기 제 2 디코더 트랜스포머 레이어의 패치 병합 해제 엔진에 의해, 상기 복수의 병합 해제된 패치들을 세그먼트화하는 단계; 및

상기 패치 병합 해제 엔진의 출력을 상기 복수의 디코더 트랜스포머 레이어들 중 제 3 디코더 트랜스포머 레이어에 제공하는 단계를 더 포함하는, 미디어 데이터를 프로세싱하는 방법.

청구항 36

제 33 항에 있어서,

상기 복수의 병합 해제된 패치들 중 각각의 병합 해제된 패치는 균일한 패치 사이즈를 갖고 상기 패치 병합 해제 엔진은 2의 패치 사이즈 감소 팩터를 적용하는, 미디어 데이터를 프로세싱하는 방법.

청구항 37

제 33 항에 있어서,

상기 복수의 병합 해제된 패치들로 상기 하나 이상의 패치들 중 각각의 패치를 세그먼트화하는 단계는 상기 복수의 병합 해제된 패치들의 특징 차원(feature dimension)을 감소시키는 것을 더 포함하는, 미디어 데이터를 프로세싱하는 방법.

청구항 38

제 32 항에 있어서,

상기 복수의 디코더 트랜스포머 레이어들에 의해, 입력으로서 상기 인코딩된 이미지 데이터의 프레임의 잠재 표현을 수신하고 비선형 변환을 적용하여 디코딩된 이미지 데이터의 프레임을 생성하는 단계를 더 포함하는, 미디어 데이터를 프로세싱하는 방법.

청구항 39

제 38 항에 있어서,

상기 비선형 변환은 합성 변환이고, 상기 디코딩된 이미지 데이터의 프레임은 상기 인코딩된 이미지 데이터의 프레임과 연관된 입력 이미지의 재구성인, 미디어 데이터를 프로세싱하는 방법.

청구항 40

제 32 항에 있어서,

레이트-왜곡(rate-distortion)에 적어도 부분적으로 기초한 손실 함수를 사용하여 상기 복수의 디코더 트랜스포머 레이어들 중 하나 이상의 디코더 트랜스포머 레이어들을 훈련하는 단계를 더 포함하는, 미디어 데이터를 프로세싱하는 방법.

청구항 41

제 40 항에 있어서,

상기 손실 함수는 레이트 왜곡을 위한 라그랑지 승수(Lagrangian multiplier)를 포함하는, 미디어 데이터를 프로세싱하는 방법.

청구항 42

제 32 항에 있어서,

상기 적어도 하나의 디코더 트랜스포머 레이어에 포함된 상기 하나 이상의 트랜스포머 블록들 중 적어도 일부는 동일한 아키텍처를 갖는, 미디어 데이터를 프로세싱하는 방법.

청구항 43

제 32 항에 있어서,

상기 적어도 하나의 디코더 트랜스포머 레이어에 포함된 상기 하나 이상의 트랜스포머 블록들의 각각은 동일한 아키텍처를 갖는, 미디어 데이터를 프로세싱하는 방법.

청구항 44

제 32 항에 있어서,

상기 인코딩된 이미지 데이터의 프레임은 인코딩된 정지 이미지를 포함하는, 미디어 데이터를 프로세싱하는 방법.

청구항 45

제 32 항에 있어서,

상기 인코딩된 이미지 데이터의 프레임은 인코딩된 비디오 프레임을 포함하는, 미디어 데이터를 프로세싱하는 방법.

청구항 46

제 32 항에 있어서,

적어도 제 1 훈련 데이터 세트 및 제 2 훈련 데이터 세트로 상기 복수의 디코더 트랜스포머 레이어들을 훈련하는 단계를 더 포함하고, 상기 제 2 훈련 데이터 세트의 데이터는 상기 제 1 훈련 데이터 세트의 데이터와 비교하여 역 시간 순서(reversed temporal order)를 갖는, 미디어 데이터를 프로세싱하는 방법.

청구항 47

제 32 항에 있어서,

상기 복수의 디코더 트랜스포머 레이어들은 일련의 연속적인 디코더 트랜스포머 레이어들을 포함하는, 미디어 데이터를 프로세싱하는 방법.

청구항 48

미디어 데이터를 프로세싱하는 방법으로서,

프레임을 복수의 패치들로 세그먼트화하는 단계; 및

입력으로서 상기 복수의 패치들을 사용하여 인코더 서브네트워크의 복수의 인코더 트랜스포머 레이어들에 의해, 인코딩된 이미지 데이터의 프레임을 생성하는 단계

를 포함하는, 미디어 데이터를 프로세싱하는 방법.

청구항 49

제 48 항에 있어서,

상기 인코딩된 이미지 데이터의 프레임을 생성하는 단계는

상기 복수의 인코더 트랜스포머 레이어들 중 제 1 인코더 트랜스포머 레이어의 제 1 트랜스포머 블록에 의해, 하나 이상의 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것;

상기 제 1 인코더 트랜스포머 레이어의 제 2 트랜스포머 블록에 의해, 하나 이상의 시프트된 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것으로서, 상기 하나 이상의 시프트된 윈도우 파티션들은 상기 하나 이상의 윈도우 파티션들과 중첩하는, 상기 셀프 어텐션을 결정하는 것;

상기 제 1 트랜스포머 블록 및 상기 제 2 트랜스포머 블록 중 하나 이상에 의해, 상기 세그먼트화된 프레임에 비선형 변환을 적용하기 위한 특징들의 하나 이상의 패치들을 결정하는 것; 및

패치 병합 엔진에 의해, 상기 제 1 인코더 트랜스포머 레이어와 제 2 인코더 트랜스포머 레이어 사이의 패치 사이즈를 증가시키는 것을 포함하는, 미디어 데이터를 프로세싱하는 방법.

발명의 설명

기술 분야

[0001] 본 개시는 일반적으로 이미지들 및/또는 비디오의 인코딩 (또는 압축) 및 디코딩 (압축해제) 을 포함하는 이미지 및 비디오 코딩에 관한 것이다. 예를 들어, 본 개시의 양태들은 시프트된 셀프 어텐션 윈도우를 갖는 트랜스포머 레이어를 사용하여 변환 코딩 및 비선형 변환을 수행하기 위한 기술에 관한 것이다.

배경 기술

[0002] 많은 디바이스들 및 시스템들은 비디오 데이터가 프로세싱되고 소비를 위해 출력될 수 있게 한다. 디지털 비디오 데이터는 소비자 및 비디오 제공자의 수요를 충족시키기 위해 대량의 데이터를 포함한다. 예를 들어, 비디오 데이터의 소비자들은, 높은 충실도, 해상도, 프레임 레이트들 등을 포함하는, 높은 품질의 비디오를 원한다. 결과적으로, 이들 수요를 충족시키기 위해 필요한 대량의 비디오 데이터는 통신 네트워크 및 그 비디오 데이터를 프로세싱하고 저장하는 디바이스에 부담을 지운다.

[0003] 비디오 데이터를 압축하기 위해 비디오 코딩 기법이 사용될 수도 있다. 비디오 코딩의 목표는 비디오 품질에 대한 열화를 회피 또는 최소화하면서 더 낮은 비트 레이트를 이용하는 형태로 비디오 데이터를 압축하는 것이다. 계속 진화하는 비디오 서비스가 이용가능하게 됨에 따라, 우수한 코딩 효율을 갖는 인코딩 기법이 필요하다.

발명의 내용

[0004] 일부 예에서, 트랜스포머 기반 신경망 아키텍처(transformer-based neural network architecture)를 사용하여 미디어 데이터를 코딩(예: 인코딩 및/또는 디코딩)하기 위한 시스템 및 기법이 설명된다. 적어도 하나의 예시적 예에 따르면, 미디어 데이터를 프로세싱하기 위한 방법으로서, 인코딩된 이미지 데이터의 프레임의 잠재 표현(latent representation)을 획득하는 단계; 및 입력으로서 인코딩된 이미지 데이터의 프레임의 잠재 표현을 사용하여 디코더 서브네트워크의 복수의 디코더 트랜스포머 레이어들에 의해, 디코딩된 이미지 데이터의 프레임을 생성하는 단계를 포함하고, 여기서 복수의 디코더 트랜스포머 레이어들 중 적어도 하나의 디코더 트랜스포머 레이어는: 특징들의 하나 이상의 패치들을 생성하고 하나 이상의 패치들 상에 적용된 하나 이상의 윈도우 파티션들 및 시프트된 윈도우 파티션들 내에서 국부적으로 셀프 어텐션(self-attention)을 결정하기 위한 하나 이상의 트랜스포머 블록들; 및 하나 이상의 패치들 중 각각의 패치의 개개의 사이즈를 줄이기 위한 패치 병합 해제 엔진(patch un-merging engine)을 포함하는, 미디어 데이터를 프로세싱하기 위한 방법이 제공된다.

[0005] 다른 예에서, 적어도 하나의 메모리 (예를 들어, 가상 콘텐츠 데이터, 하나 이상의 이미지들 등과 같은 데이터를 저장하도록 구성됨) 및 적어도 하나의 메모리에 커플링된 하나 이상의 프로세서들 (예를 들어, 회로에 구현됨) 을 포함하는, 미디어 데이터를 프로세싱하기 위한 장치가 제공된다. 하나 이상의 프로세서는 인코딩된 이미지 데이터의 프레임의 잠재 표현(latent representation)을 획득하고; 입력으로서 인코딩된 이미지 데이터의 프레임의 잠재 표현을 사용하여 디코더 서브네트워크의 복수의 디코더 트랜스포머 레이어들에 기초하여, 디코딩된 이미지 데이터의 프레임을 생성하도록 구성되고 이를 할 수 있고, 여기서 복수의 디코더 트랜스포머 레이어들 중 적어도 하나의 디코더 트랜스포머 레이어는: 특징들의 하나 이상의 패치들을 생성하고 하나 이상의 패치들 상에 적용된 하나 이상의 윈도우 파티션들 및 시프트된 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하도록 구성된 하나 이상의 트랜스포머 블록들; 및 하나 이상의 패치들 중 각각의 패치의 개개의 사이즈를 줄이도록 구성되는 패치 병합 해제 엔진을 포함한다.

- [0006] 다른 예에서, 명령들을 저장한 비밀시적 컴퓨터 판독 가능 매체로서, 상기 명령들은 하나 이상의 프로세서들에 의해 실행될 때 상기 하나 이상의 프로세서들로 하여금: 인코딩된 이미지 데이터의 프레임의 잠재 표현(latent representation)을 획득하게 하고; 입력으로서 인코딩된 이미지 데이터의 프레임의 잠재 표현을 사용하여 디코더 서브네트워크의 복수의 디코더 트랜스포머 레이어들에 기초하여, 디코딩된 이미지 데이터의 프레임을 생성하게 하고, 여기서 복수의 디코더 트랜스포머 레이어들 중 적어도 하나의 디코더 트랜스포머 레이어는: 특징들의 하나 이상의 패치들을 생성하고 하나 이상의 패치들 상에 적용된 하나 이상의 윈도우 파티션들 및 시프트된 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하도록 구성된 하나 이상의 트랜스포머 블록들; 및 하나 이상의 패치들 중 각각의 패치의 개개의 사이즈를 줄이도록 구성되는 패치 병합 해제 엔진을 포함하는, 비밀시적 컴퓨터 판독 가능 매체가 제공된다.
- [0007] 미디어 데이터를 프로세싱하기 위한 장치가 제공된다. 장치는: 인코딩된 이미지 데이터의 프레임의 잠재 표현(latent representation)을 획득하는 수단; 및 입력으로서 인코딩된 이미지 데이터의 프레임의 잠재 표현을 사용하여 디코더 서브네트워크의 복수의 디코더 트랜스포머 레이어들에 기초하여, 디코딩된 이미지 데이터의 프레임을 생성하는 수단을 포함하고, 여기서 복수의 디코더 트랜스포머 레이어들 중 적어도 하나의 디코더 트랜스포머 레이어는: 특징들의 하나 이상의 패치들을 생성하고 하나 이상의 패치들 상에 적용된 하나 이상의 윈도우 파티션들 및 시프트된 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하도록 구성된 하나 이상의 트랜스포머 블록들; 및 하나 이상의 패치들 중 각각의 패치의 개개의 사이즈를 줄이도록 구성되는 패치 병합 해제 엔진을 포함한다.
- [0008] 일부 양태에서, 디코딩된 이미지 데이터의 프레임을 생성하기 위해, 위에 설명된 방법, 장치 및 컴퓨터 판독 가능 매체는: 복수의 디코더 트랜스포머 레이어들 중 제 1 디코더 트랜스포머 레이어의 제 1 트랜스포머 블록에 의해, 하나 이상의 패치들 상에 적용된 하나 이상의 제 1 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것; 제 1 디코더 트랜스포머 레이어의 제 2 트랜스포머 블록에 의해, 하나 이상의 패치들 상에 적용된 하나 이상의 제 2 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것으로서, 하나 이상의 제 2 윈도우 파티션들은 하나 이상의 제 1 윈도우 파티션들 중 인접한 것들 사이에 하나 이상의 경계들과 중첩되도록 시프트되는, 상기 셀프 어텐션을 결정하는 것; 및 패치 병합 해제 엔진에 의해, 복수의 병합 해제된 패치들로 하나 이상의 패치들의 각각의 패치를 세그먼트화하는 것으로서, 복수의 병합 해제된 패치들은 중첩되지 않는, 상기 하나 이상의 패치들의 각각의 패치를 세그먼트화하는 것을 포함할 수 있다.
- [0009] 일부 양태들에서, 위에 설명된 방법, 장치 및 컴퓨터 판독 가능 매체는: 복수의 디코더 트랜스포머 레이어들 중 제 2 디코더 트랜스포머 레이어의 제 1 트랜스포머 블록에 복수의 병합 해제된 패치들을 제공하는 것을 포함할 수 있다.
- [0010] 일부 양태들에서, 위에 설명된 방법, 장치 및 컴퓨터 판독 가능 매체는: 제 2 디코더 트랜스포머 레이어의 패치 병합 해제 엔진에 의해, 복수의 병합 해제된 패치들을 세그먼트화하는 것; 및 패치 병합 해제 엔진의 출력을 복수의 디코더 트랜스포머 레이어들 중 제 3 디코더 트랜스포머 레이어에 제공하는 것을 포함할 수 있다.
- [0011] 일부 양태에서, 복수의 병합 해제된 패치들 중 각각의 병합 해제된 패치는 균일한 패치 사이즈를 갖고 패치 병합 해제 엔진은 2의 패치 사이즈 감소 팩터(factor)를 적용한다.
- [0012] 일부 양태에서, 복수의 병합 해제된 패치들로 하나 이상의 패치들 중 각각의 패치를 세그먼트화하기 위해, 위에 설명된 방법, 장치 및 컴퓨터 판독 가능 매체는 복수의 병합 해제된 패치들의 특징 차원(feature dimension)를 감소시키는 것을 포함할 수 있다.
- [0013] 일부 양태들에서, 위에 설명된 방법, 장치 및 컴퓨터 판독 가능 매체는: 복수의 디코더 트랜스포머 레이어들에 의해, 인코딩된 이미지 데이터의 프레임의 잠재 표현을 입력으로 수신하고 비선형 변환을 적용하여 디코딩된 이미지 데이터의 프레임을 생성하는 것을 포함할 수 있다.
- [0014] 일부 양태들에서, 비선형 변환은 합성 변환이고, 디코딩된 이미지 데이터의 프레임은 인코딩된 이미지 데이터의 프레임과 연관된 입력 이미지의 재구성이다.
- [0015] 일부 양태들에서, 위에 설명된 방법, 장치 및 컴퓨터 판독 가능 매체는: 레이트-왜곡(rate-distortion)에 적어도 부분적으로 기초한 손실 함수를 사용하여 복수의 디코더 트랜스포머 레이어들 중 하나 이상의 디코더 트랜스포머 레이어들을 훈련하는 것을 포함할 수 있다. 일부 경우들에서, 손실 함수는 레이트 왜곡을 위한 라그랑지 승수(Lagrangian multiplier)를 포함한다.

- [0016] 일부 양태들에서, 적어도 하나의 디코더 트랜스포머 레이어에 포함된 하나 이상의 트랜스포머 블록들 중 적어도 일부는 동일한 아키텍처를 갖는다.
- [0017] 일부 양태들에서, 적어도 하나의 디코더 트랜스포머 레이어에 포함된 하나 이상의 트랜스포머 블록들 각각은 동일한 아키텍처를 갖는다.
- [0018] 일부 양태들에서, 인코딩된 이미지 데이터의 프레임은 인코딩된 정지 이미지를 포함한다.
- [0019] 일부 양태들에서, 인코딩된 이미지 데이터의 프레임은 인코딩된 비디오 프레임을 포함한다.
- [0020] 일부 양태들에서, 위에 설명된 방법, 장치 및 컴퓨터 판독 가능 매체는: 적어도 제 1 훈련 데이터 세트 및 제 2 훈련 데이터 세트로 복수의 디코더 트랜스포머 레이어들을 훈련하는 것을 포함할 수 있다, 제 2 훈련 데이터 세트의 데이터는 제 1 훈련 데이터 세트의 데이터와 비교하여 역 시간 순서(reversed temporal order)를 갖는다.
- [0021] 일부 양태에서, 복수의 디코더 트랜스포머 레이어들은 일련의 연속적인 디코더 트랜스포머 레이어들을 포함한다.
- [0022] 다른 예시적 예에 따르면, 미디어 데이터를 프로세싱하기 위한 방법이 제공된다: 그 방법은 프레임을 복수의 패치들로 세그먼트화하는 단계; 및 입력으로서 복수의 패치들을 사용하여 인코더 서브네트워크의 복수의 인코더 트랜스포머 레이어들에 의해, 인코딩된 이미지 데이터의 프레임을 생성하는 단계를 포함한다.
- [0023] 다른 예에서, 적어도 하나의 메모리 (예를 들어, 가상 콘텐츠 데이터, 하나 이상의 이미지들 등과 같은 데이터를 저장하도록 구성됨) 및 적어도 하나의 메모리에 커플링된 하나 이상의 프로세서들 (예를 들어, 회로에 구현됨) 을 포함하는, 미디어 데이터를 프로세싱하기 위한 장치가 제공된다. 하나 이상의 프로세서들은 프레임을 복수의 패치들로 세그먼트화하고; 입력으로서 복수의 패치들을 사용하여 인코더 서브네트워크의 복수의 인코더 트랜스포머 레이어들에 기초하여, 인코딩된 이미지 데이터의 프레임을 생성하도록 구성되고 이를 할 수 있다.
- [0024] 다른 예에서, 명령들을 저장한 비일시적 컴퓨터 판독 가능 매체로서, 상기 명령들은, 하나 이상의 프로세서들에 의해 실행될 때 상기 하나 이상의 프로세서들로 하여금 프레임을 복수의 패치들로 세그먼트화하고; 입력으로서 복수의 패치들을 사용하여 인코더 서브네트워크의 복수의 인코더 트랜스포머 레이어들에 기초하여, 인코딩된 이미지 데이터의 프레임을 생성하게 하는, 비일시적 컴퓨터 판독 가능 매체가 제공된다.
- [0025] 미디어 데이터를 프로세싱하기 위한 장치가 제공된다. 장치는 프레임을 복수의 패치들로 세그먼트화하는 수단; 및 입력으로서 복수의 패치들을 사용하여 인코더 서브네트워크의 복수의 인코더 트랜스포머 레이어들에 기초하여, 인코딩된 이미지 데이터의 프레임을 생성하는 수단을 포함한다.
- [0026] 일부 양태에서, 인코딩된 이미지 데이터의 프레임을 생성하기 위해, 위에 설명된 방법, 장치 및 컴퓨터 판독 가능 매체는 복수의 인코더 트랜스포머 레이어들 중 제 1 인코더 트랜스포머 레이어의 제 1 트랜스포머 블록에 의해, 하나 이상의 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것; 제 1 인코더 트랜스포머 레이어의 제 2 트랜스포머 블록에 의해, 하나 이상의 시프트된 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것으로서, 하나 이상의 시프트된 윈도우 파티션들은 하나 이상의 윈도우 파티션들과 중첩하는, 상기 셀프 어텐션을 결정하는 것; 제 1 트랜스포머 블록 및 제 2 트랜스포머 블록 중 하나 이상에 의해, 세그먼트화된 프레임에 비선형 변환을 적용하기 위한 특징들의 하나 이상의 패치들을 결정하는 것; 및 패치 병합 엔진에 의해, 제 1 인코더 트랜스포머 레이어와 제 2 인코더 트랜스포머 레이어 사이의 패치 사이즈를 증가시키는 것을 포함할 수 있다.
- [0027] 일부 양태들에서, 패치 병합 엔진은 제 1 인코더 트랜스포머 레이어로부터의 복수의 인접한 패치들을 제 2 인코더 트랜스포머 레이어에 제공되는 병합된 패치로 결합하도록 구성된다.
- [0028] 일부 양태들에서, 제 1 인코더 트랜스포머 레이어의 제 2 트랜스포머 블록의 출력은 제 2 인코더 트랜스포머 레이어의 입력에 커플링된다.
- [0029] 일부 양태들에서, 위에 설명된 방법, 장치 및 컴퓨터 판독 가능 매체는 입력으로서 복수의 패치들을 사용하여 인코더 서브네트워크의 복수의 인코더 트랜스포머 레이어들에 의해, 세그먼트화된 프레임에 대한 계층적 특징 맵을 생성하는 것; 및 계층적 특징 맵으로부터 인코딩된 이미지 데이터의 프레임을 생성하는 것을 포함할 수 있다.
- [0030] 일부 양태에서, 복수의 패치들 중 각각의 패치는 균일한 사이즈를 가지며 세그먼트화된 프레임의 하나 이상의

픽셀들을 포함한다.

- [0031] 일부 양태들에서, 위에 설명된 방법, 장치 및 컴퓨터 판독 가능 매체는 패치 병합 엔진을 사용하여, 인접한 패치들의 하나 이상의 서브세트들에서 얻은 특징들을 연쇄(concatenating)시켜 패치 사이즈를 늘리는 것을 포함할 수 있고, 인접한 패치들의 각각의 서브세트는 패치 병합 엔진에 의해 병합된 패치 출력으로 병합된다.
- [0032] 일부 양태들에서, 제 1 트랜스포머 블록 및 제 2 트랜스포머 블록은 동일한 아키텍처를 갖는다.
- [0033] 일부 양태들에서, 위에 설명된 방법, 장치 및 컴퓨터 판독 가능 매체는 제 1 인코더 트랜스포머 레이어 이전에 인코더 서브네트워크의 선형 임베딩 레이어에 복수의 패치들을 제공하는 것을 포함할 수 있다.
- [0034] 일부 양태들에서, 인코딩된 이미지 데이터의 프레임은 이미지 데이터의 잠재 표현이다.
- [0035] 일부 양태들에서, 잠재 표현은 인코더 서브네트워크의 복수의 인코더 트랜스포머 레이어들에 의해 생성된 계층적 특징 맵이다.
- [0036] 일부 양태들에서, 위에 설명된 방법, 장치 및 컴퓨터 판독 가능 매체는 레이트-왜곡 손실에 기초한 손실 함수를 사용하여 복수의 인코더 트랜스포머 레이어들 중 하나 이상의 인코더 트랜스포머 레이어들을 훈련하는 것을 포함할 수 있다. 일부 경우들에서, 손실 함수는 레이트 왜곡을 위한 라그랑지 승수(Lagrangian multiplier)를 포함한다.
- [0037] 일부 양태에서, 복수의 패치들은 정지 이미지 프레임 또는 비디오 프레임을 포함하는 입력으로부터 세그먼트화된다.
- [0038] 일부 양태들에서, 위에 설명된 방법, 장치 및 컴퓨터 판독 가능 매체는 팩터라이즈드 프라이어(factorized prior)를 사용하여 인코딩된 이미지 데이터를 엔트로피 코딩하는 것을 포함할 수 있다.
- [0039] 일부 양태들에서, 장치는 모바일 디바이스 (예컨대, 모바일 전화 또는 소위 "스마트 폰", 태블릿 컴퓨터, 또는 다른 유형의 모바일 디바이스), 웨어러블 디바이스, 확장 현실 디바이스 (예컨대, 가상 현실 (VR) 디바이스, 증강 현실 (AR) 디바이스, 또는 혼합 현실 (MR) 디바이스), 개인용 컴퓨터, 랩탑 컴퓨터, 비디오 서버, 텔레비전, 차량 (또는 차량의 컴퓨팅 디바이스) 또는 다른 디바이스를 포함한다. 일부 양태들에서, 장치는 하나 이상의 이미지들 또는 비디오 프레임들을 캡처하기 위한 적어도 하나의 카메라를 포함한다. 예를 들어, 장치는 비디오 프레임들을 포함하는 하나 이상의 이미지들 및/또는 하나 이상의 비디오들을 캡처하기 위한 카메라 (예를 들어, RGB 카메라) 또는 다수의 카메라들을 포함할 수 있다. 일부 양태들에서, 장치는 하나 이상의 이미지들, 비디오, 통지 또는 다른 디스플레이가능한 데이터를 디스플레이하기 위한 디스플레이를 포함한다. 일부 양태들에서, 장치는 하나 이상의 비디오 프레임 및/또는 선택스 데이터를 송신 매체를 통해 적어도 하나의 디바이스에 송신하도록 구성된 송신기를 포함한다. 일부 양태들에서, 프로세서는 뉴럴 프로세싱 유닛(neural processing unit; NPU), 중앙 프로세싱 유닛(central processing unit; CPU), 그래픽 프로세싱 유닛(graphics processing unit; GPU) 또는 다른 프로세싱 디바이스 또는 컴포넌트를 포함한다.
- [0040] 이 개요는, 청구된 요지의 핵심적인 또는 본질적인 특징들을 식별하도록 의도되지 않으며, 청구된 요지의 범위를 결정하는 데 별개로 사용되도록 의도되지도 않는다. 청구물은 본 특허의 전체 명세서, 임의의 또는 모든 도면들, 및 각각의 청구항의 적절한 부분들을 참조하여 이해되어야 한다.
- [0041] 기술한 내용은, 다른 특징들 및 실시형태들과 함께, 다음의 명세서, 청구항들 및 첨부 도면들을 참조하면 더욱 명백해질 것이다.

도면의 간단한 설명

- [0042] 본 출원의 예시적 실시형태들은 다음 도면들을 참조하여 이하에서 상세히 설명된다.
- 도 1 은 시스템-온-칩(SOC)의 예시적인 구현을 예시한다;
- 도 2a 는 전체적으로 연결된 신경망의 예를 예시한다;
- 도 2b 는 국부적으로 연결된 신경망의 예를 예시한다;
- 도 3 은 일부 예들에 따른, 신경망 기반 시스템을 사용하여 이미지 및/또는 비디오 코딩 (인코딩 및 디코딩) 을 수행하도록 동작가능한 디바이스를 포함하는 시스템의 예를 예시하는 도면이다;
- 도 4 는 일부 예들에 따른, 적색-녹색-청색(RGB) 포맷을 갖는 입력을 위한 엔드-투-엔드 신경망 기반 이미지 및

비디오 코딩 시스템의 예를 예시하는 도면이다;

도 5a 는 일부 예들에 따른, 신경망 기반 이미지 및 비디오 코딩 시스템의 인코더를 위한 트랜스포머 기반 신경망 아키텍처의 예를 예시하는 도면이다;

도 5b 는 일부 예들에 따른, 신경망 기반 이미지 및 비디오 코딩 시스템의 디코더를 위한 트랜스포머 기반 신경망 아키텍처의 예를 예시하는 도면이다;

도 5c 는 일부 예들에 따른, 신경망 기반 이미지 및 비디오 코딩 시스템을 위한 트랜스포머 기반 엔드-투-엔드 신경망 아키텍처의 예를 예시하는 도면이다;

도 6a 는 일부 예들에 따른, 시프트된 윈도우 트랜스포머 블록들 쌍의 예시적인 아키텍처를 예시하는 도면이다;

도 6b는 일부 예들에 따른, 하나 이상의 트랜스포머 기반 신경망 아키텍처를 사용하는 비디오 코딩 시스템의 예를 예시하는 도면이다;

도 7a는 일부 예들에 따른, 인코더 또는 디코더 신경망 기반 이미지 및 비디오 코딩 시스템의 트랜스포머 레이어들 사이에 적용될 수 있는 패치 병합 또는 패치 병합 해제 프로세스의 예를 예시하는 도면이다;

도 7b는 일부 예들에 따른, 인코더 또는 디코더 신경망 기반 이미지 및 비디오 코딩 시스템의 2개의 셀프 어텐션 레이어들 사이의 시프트된 윈도우 셀프 어텐션 산출의 예를 예시하는 도면이다;

도 8 은 일부 예들에 따른, 이미지 및/또는 비디오 데이터를 프로세싱하기 위한 프로세스의 예를 예시하는 플로우차트이다;

도 9 는 일부 예들에 따른, 이미지 및/또는 비디오 데이터를 프로세싱하기 위한 프로세스의 다른 예를 예시하는 플로우차트이다; 그리고

도 10 는 본 명세서에 설명된 다양한 기법을 구현할 수 있는 예시적인 컴퓨팅 디바이스의 예시적인 컴퓨팅 디바이스 아키텍처를 예시한다.

발명을 실시하기 위한 구체적인 내용

[0043]

상세한 설명

[0044]

본 개시의 특정 양태들 및 실시형태들이 이하에 제공된다. 이들 양태들 및 실시형태들 중 일부는 독립적으로 적용될 수 있고 그들 중 일부는 당업자에게 자명한 바와 같이 조합하여 적용될 수도 있다. 다음의 설명에 있어서, 설명의 목적으로, 특정 상세들이 본 출원의 실시형태들의 철저한 이해를 제공하기 위해 제시된다. 하지만, 다양한 실시형태들이 이들 특정 상세 없이도 실시될 수도 있음이 분명할 것이다. 도면들 및 설명은 제한적인 것으로 의도되지 않는다.

[0045]

다음의 설명은 오직 예시적인 실시형태들을 제공할 뿐이고, 본 개시의 범위, 적용가능성, 또는 구성을 한정하도록 의도되지 않는다. 오히려, 예시적인 실시형태들의 설명은 예시적인 실시형태를 구현하기 위한 가능한 설명을 당업자에게 제공할 것이다. 첨부된 청구항들에 제시된 바와 같은 본 출원의 사상 및 범위로부터 벗어남이 없이 엘리먼트들의 기능 및 배열에서 다양한 변경들이 이루어질 수도 있음이 이해되어야 한다.

[0046]

디지털 비디오 데이터는 특히 고품질 비디오 데이터에 대한 수요가 계속 증가함에 따라 대량의 데이터를 포함할 수 있다. 예를 들어, 비디오 데이터의 소비자들은 통상적으로, 높은 충실도, 해상도, 프레임 레이트들 등을 갖는 점점 더 높은 품질의 비디오를 원한다. 그러나, 높은 수요를 충족시키는데 필요한 대량의 비디오 데이터는 종종 큰 대역폭 및 저장 필요와 연관되어, 비디오 데이터를 프로세싱하고 저장하는 디바이스들에 대해서뿐만 아니라 통신 네트워크들에 대해 상당한 부담을 지운다.

[0047]

비디오 데이터를 코딩하기 위해 다양한 기법들이 사용될 수 있다. 예를 들어, 비디오 코딩은 특정 비디오 코딩 표준에 따라 수행될 수 있다. 비디오 코딩 표준의 예로는 MPEG(moving picture experts group)에 의해 개발된 HEVC(high-efficiency video coding), AVC(advanced video coding), , 및 VVC(versatile video coding), 그리고 AOM(Alliance for Open Media)에 의해 개발된 AV1(AOMedia Video 1)가 있다. 비디오 코딩은 종종, 비디오 이미지들 또는 시퀀스들에 존재하는 리던던시(redundancy)를 이용하는 인터-예측(inter-prediction), 인트라-예측(intra-prediction)과 같은 예측 방법들을 사용한다. 비디오 코딩 기법들의 공통 목적은, 예를 들어, 압축된 비디오 데이터가 압축 해제될 때, 비디오 품질의 열화를 회피 또는 최소화하면서, 더 낮은 비트 레이트를 사용하는 형태로 비디오 데이터를 압축하는 것이다. 비디오 서비스에 대한 수요가 증가하고 새로운 비디오 서비스

가 이용가능함에 따라, 더 나은 코딩 효율, 성능 및 레이트 제어를 갖춘 코딩 기술이 필요하다.

[0048] 하나 이상의 머신 학습(ML) 시스템들을 사용하여 이미지 및/또는 비디오 코딩을 수행하기 위한 시스템들, 장치들, 프로세스들(방법들이라고도 지칭됨), 및 컴퓨터 판독가능 매체들(집합적으로 "시스템들 및 기법들"이라고 지칭됨)이 본 명세서에 설명된다. 일반적으로 ML은 인공 지능(AI)의 서브세트로 고려될 수 있다. ML 시스템들은, 컴퓨터 시스템들이 명시적 명령들의 사용 없이, 패턴들 및 추론에 의존하는 것에 의해 다양한 태스크들을 수행하기 위해 사용할 수 있는 알고리즘들 및 통계 모델들을 포함할 수 있다. ML 시스템의 하나의 예는 인공 뉴런들(예를 들어, 뉴런 모델들)의 상호연결된 그룹을 포함할 수 있는 신경망(인공 신경망으로도 지칭됨)이다. 신경망들은 특히 이미지 및/또는 비디오 코딩, 이미지 분석 및/또는 컴퓨터 비전 애플리케이션들, 인터넷 프로토콜(IP) 카메라들, 사물 인터넷(IoT) 디바이스들, 자율 차량들, 서비스 로봇들과 같은 다양한 애플리케이션들 및/또는 디바이스들을 위해 사용될 수 있다.

[0049] 신경망 내의 개별 노드들은 입력 데이터를 취하고 데이터에 대해 단순한 연산들을 수행함으로써 생물학적 뉴런들을 에뮬레이트할 수 있다. 입력 데이터에 대해 수행된 단순한 연산들의 결과들은 다른 뉴런들에 선택적으로 전달된다. 가중 값들은 네트워크에서 각각의 벡터 및 노드와 연관되며 이들 값은 입력 데이터가 출력 데이터와 관련되는 방식을 제약한다. 예를 들어, 각각의 노드의 입력 데이터는 대응하는 가중 값들로 곱해질 수도 있고, 그 곱들은 합산될 수도 있다. 곱들의 합은 임의적 바이어스(bias)에 의해 조정될 수도 있고, 활성화 함수가 결과에 적용되어, 노드의 출력 신호 또는 "출력 활성화" (때때로 특징 맵 또는 활성화 맵으로 지칭됨)를 낳을 수도 있다. 가중 값들은 초기에 네트워크를 통한 훈련 데이터의 반복 흐름에 의해 결정될 수 있다(예를 들어, 가중 값들은 네트워크가 그들의 전형적인 입력 데이터 특성들에 의해 특정 클래스들을 식별하는 방법을 학습하는 훈련 페이즈 동안 확립된다).

[0050] 특히, 컨볼루션 신경망(CNN), 순환 신경망(RNN), 생성 적대 망(GAN), 다층 퍼셉트론(MLP) 신경망, 트랜스포머 신경망과 같은 상이한 유형의 신경망들이 존재한다. 예를 들어, 컨볼루션 신경망(CNN)은 피드-포워드 인공 신경망의 일종이다. 컨볼루션 신경망들은 각각 수용 필드(receptive field)(예를 들어, 입력 공간의 공간적으로 로컬화된 영역)를 갖고 입력 공간을 집합적으로 타일링하는 인공 뉴런들의 집합들을 포함할 수 있다. RNN은 레이어의 출력을 저장하고 레이어의 결과를 예측하는 데 도움이 되도록 이 출력을 입력에 다시 공급하는 원칙에 대해 작동한다. GAN은 입력 데이터에서 패턴들을 학습할 수 있는 생성 신경망의 형태이며, 따라서 신경망 모델은 원래 데이터셋으로부터 합리적으로 얻어질 수 있었던 새로운 합성 출력들을 생성할 수 있다. GAN은, 합성된 출력을 생성하는 생성 신경망 및 진본성에 대해 출력을 평가하는 판별 신경망을 포함한, 함께 동작하는 2개의 신경망들을 포함할 수 있다. MLP 신경망들에서, 데이터는 입력 레이어에 공급될 수도 있고, 하나 이상의 은닉 레이어들은 데이터에 대한 추상화의 레벨들을 제공한다. 그런 다음, 추상화된 데이터에 기초하여 출력 레이어 상에서 예측들이 이루어질 수도 있다.

[0051] 딥 러닝(DL)은 머신 학습 기술의 하나의 예이며 ML의 서브세트로 간주될 수 있다. 많은 DL 접근법들은 RNN 또는 CNN과 같은 신경망을 기반으로 하며 여러 레이어들을 활용한다. 심층 신경망에서 다수의 레이어들의 사용은 주어진 원시 데이터 입력으로부터 점진적으로 더 높은 수준의 특징들이 추출되는 것을 허락할 수 있다. 예를 들어, 인공 뉴런들의 제 1 레이어의 출력은 인공 뉴런들의 제 2 레이어에 대한 입력이 되고, 인공 뉴런들의 제 2 레이어의 출력은 인공 뉴런들의 제 3 레이어에 대한 입력이 되는 등등이다. 전체 심층 신경망의 입력과 출력 사이에 위치한 레이어들은 종종 히든 레이어(Hidden Layer)들이라고 부른다. 히든 레이어들은, 최종 또는 원하는 표현이 심층 신경망의 최종 출력으로 얻어질 때까지, 선행 레이어로부터의 중간 입력을 후속 레이어에 제공될 수 있는 약간 더 추상적이고 복잡한 표현으로 변환하는 법을 학습한다(예를 들어, 훈련된다).

[0052] CNN(및 일부 경우에 RNN)은 컴퓨터 비전을 구현하거나 또는 이미지/비디오 코딩을 수행하는 것과 같은 이미지 기반 입력들의 맥락에서 일반적으로 사용된다. 예를 들어, 이미지/비디오 코딩의 맥락에서, 기존 접근법들은 오토그래시브 CNN을 사용하여 엔트로피 인코딩 및 디코딩을 수행하는 데 사용되는 프라이어 모델(prior model)을 구현한다. 일부 기존 접근법들은 전통적인 코덱과 동등하거나 이를 능가하는 레이트 왜곡 성능을 달성하는 것으로 나타났지만, 오토그래시브 CNN에 의해 수행되는 오토그래시브 디코딩의 레이턴시와 복잡성으로 인해 낮은 레이턴시(예: 실시간 또는 근 실시간) 코딩을 달성할 수 없다.

[0053] 낮은 레이턴시로 이미지 및 비디오 코딩을 정확하게 그리고 보다 효율적으로 수행하기 위한 시스템과 기술이 필요하다. ML 및 DL 기반 접근법들은 레이트 왜곡 성능 측면에서 기존 코덱에 비해 이론적인 개선을 보여주었지만, 이들 기존 접근법들은 (예: 위에 언급된 바처럼, 오토그래시브 CNN에 의해 수행되는 오토그래시브 디코딩에 기반하여) 낮은 레이턴시로 실행할 수 없다는 점으로 인해 종종 제한된다. 더욱이, 이미지와 비디오

오 데이터의 해상도가 계속 증가함에 따라, 이러한 기존 접근법들의 계산 런타임도 계속해서 증가하기만 할 것 같다.

[0054] 일부 양태들에서, 본 명세서에 설명된 시스템 및 기법은, CNN 기반 변환과 같은 다른 신경망 기반 이미지 및/또는 비디오 코딩 시스템보다 더 빠른 코딩(예를 들어 디코딩)으로 낮은 레이턴시 이미지 및/또는 비디오 코딩을 수행할 수 있는 트랜스포머 기반 이미지 및 비디오 코딩 시스템을 포함한다. 예를 들어, 트랜스포머 기반 변환을 사용하여, 본 명세서에 설명된 시스템과 기법들은 Kodak 이미지 압축 데이터세트에서 적어도 7%, 그리고 UVG(Ultra Video Group) 비디오 압축 데이터세트 (저 지연 모드)에서 12%의 코딩 효율성 이득을 달성할 수 있다.

[0055] 트랜스포머는 어텐션 메커니즘(attention mechanism)을 활용하여 입력 데이터의 각각의 부분의 유의성(significance)을 차등적으로 가중화하고 장범위 종속성(long-range dependency)을 모델링하는 일종의 딥 러닝 모델이다. 트랜스포머는 순차 입력 데이터를 처리하는 데 종종 사용되지만, 트랜스포머는 데이터가 원래 수신되거나 배열되었던 것과 동일한 순차적 순서로 데이터를 반드시 프로세싱하는 것은 아니다. 또한, 트랜스포머는 어텐션을 사용하여 입력 데이터의 서브 부분들 간의 문맥상 관계(contextual relationship)를 결정할 수 있기 때문에, 어텐션 또는 셀프 어텐션을 계산할 때와 같이 트랜스포머는 서브 부분들의 일부 또는 전부를 병렬로 프로세싱할 수 있다. 이러한 병렬화는 예를 들어 RNN, CNN 또는 동일한 태스크를 수행하도록 훈련된 다른 신경망에 비해 더 큰 계산적 유연성을 제공할 수 있다.

[0056] 일부 경우에, 본 명세서에 설명된 트랜스포머 기반 이미지 및 비디오 코딩 시스템은 인코더 서브네트워크 및 디코더 서브네트워크를 포함하며, 각각은 다수의 연속적인 시프트된 윈도우 트랜스포머 레이어들을 포함한다. 인코더 서브 네트워크는 변환을 적용하여 입력 이미지를 잠재 표현으로 변환하는 반면, 디코더 서브 네트워크는 변환을 적용하여 잠재 표현을 입력 이미지의 재구성된 버전인 재구성된 이미지로 전환한다. 입력 이미지들에는 정지 이미지(예: 사진 및 기타 유형의 정지 이미지)와 비디오 이미지(예: 비디오의 프레임들)가 포함될 수 있다. 일부 예들에서, 인코더 및 디코더 서브네트워크는, (예를 들어, CNN 기반 접근법에서 사용되는 더 계산적으로 복잡한 오토그래피크 프라이어와 반대로) 팩터라이즈 프라이어와 함께 사용되어, 이미지 데이터의 낮은 레이턴시 인코딩 및 디코딩을 허용하는 비선형 변환을 적용할 수 있다. 더욱이, 여기에 설명된 트랜스포머 기반 이미지 및 비디오 코딩 시스템은 기존 신경망 이미지 및 비디오 코딩 시스템(예: CNN 기반 이미지 및 비디오 코딩 시스템)과 연관된 레이트 왜곡 손실과 매치하거나 이를 개선하는 레이트 왜곡 손실로 이러한 낮은 레이턴시 인코딩 및 디코딩을 달성할 수 있다.

[0057] 일부 예에서, 디코더 서브네트워크는, 인코더 서브네트워크에 포함된 패치 병합 엔진의 기능을 대체하고 반전시키는 패치 병합 해제 엔진을 제외하고는, 인코더 서브네트워크에 대칭이다. 인코더 및 디코더 서브네트워크들은 일련의 패치들(여기서는 "패치 토큰들"이라고도 함)상에서 동작한다. 일부 경우들에서, 일련의 패치들이 처음에 입력 이미지의 비중첩 세그먼트화로서 인코더 서브네트워크에 형성된다.

[0058] 일부 양태들에서, 인코더 서브 네트워크 및 디코더 서브 네트워크는 시프트된 윈도우 접근법을 갖는 수정된 셀프 어텐션 계산을 활용한다. 시프트된 윈도우 접근법은 셀프 어텐션 계산을 비중첩 로컬 윈도우들로 제한하는 한편 또한 크로스 윈도우 연결을 허용하는 셀프 어텐션 레이어들의 쌍(셀프 어텐션 레이어 쌍이라고도 함)을 기반으로 한다. 일부 예에서, 크로스 윈도우 연결은 더 깊은 셀프 어텐션 레이어에서 활용되어 각각이 하위 셀프 어텐션 레이어의 다수의 개별 어텐션 윈도우들에 걸쳐 있는(예를 들어, 이후부터의 엘리먼트들에 기초하는) 어텐션 벡터들을 결정할 수 있다. 예시적인 예에서, 셀프 어텐션 레이어 쌍 중 제 1 레이어는 제 1 파티셔닝 구성을 적용하여 다수의 패치들을 각각 포함하는 비중첩 윈도우들로 패치들의 세트를 나눌 수 있다 - 그런 다음 셀프 어텐션이 각각의 윈도우 내에서 국부적으로 계산된다. 셀프 어텐션 레이어 쌍 중 제 2 레이어에서는, 윈도우 파티셔닝이 시프트되어, 그 결과 새로운 윈도우들이 제 1 셀프 어텐션 레이어로부터의 윈도우들과 중첩하게 된다. 제 2 셀프 어텐션 레이어의 시프트된 윈도우들에서 셀프 어텐션 계산은 제 1 셀프 어텐션 레이어에서 이전 윈도우들의 경계들을 크로스하여, 그것들간에 연결을 제공한다.

[0059] 셀프 어텐션 계산을 제 1 셀프 어텐션 레이어의 비중첩 로컬 윈도우로 제한함으로써, 여기에 설명된 트랜스포머 기반 이미지 및 비디오 코딩 시스템은 낮은 레이턴시 이미지 및 비디오 인코딩 및 디코딩(그리고 일부 경우에는, 이를테면 빠른 엔트로피 모델들이 사용될 때, 실시간 또는 근 실시간 인코딩 및 디코딩)을 지원하는 더 큰 효율성과 계산적 성능을 달성할 수 있다. 예를 들어, 일부 양태들에서, 트랜스포머 기반 이미지 및 비디오 코딩 시스템은 이미지 사이즈에 상대적으로 선형적인 계산적 복잡성을 갖는다. 제 2 셀프 어텐션 레이어의 시프트된 윈도우 파티셔닝에 의해 결정된 크로스 윈도우 연결을 도입함으로써, 여기에 설명된 트랜스포머 기반

이미지 및 비디오 코딩 시스템은 CNN 기반 및 다른 기존 이미지 및 비디오 코딩 접근법들과 연관된 레이트 왜곡 손실에 대해 매치하거나 개선하는 레이트 왜곡 손실을 달성할 수 있다.

[0060] 본 개시의 다양한 양태들이 도면들을 참조하여 설명될 것이다. 도 1 은 시스템-온-칩 (SOC)(100) 의 예시적인 구현을 예시하며, 이는 본 명세서에서 설명된 기능들 중 하나 이상을 수행하도록 구성된 중앙 프로세싱 유닛 (CPU)(102) 또는 멀티-코어 CPU 를 포함할 수도 있다. 다른 정보 중에서도, 파라미터들 또는 변수들(예를 들어, 신경 신호들 및 시냅스 가중치들), 계산 디바이스와 연관된 시스템 파라미터들(예를 들어, 가중치들을 갖는 신경망), 지연들, 주파수 빈 정보, 태스크 정보는 신경 프로세싱 유닛(NPU)(108)과 연관된 메모리 블록에, CPU(102)와 연관된 메모리 블록에, 그래픽 프로세싱 유닛(GPU)(104)과 연관된 메모리 블록에, 디지털 신호 프로세서(DSP)(106)와 연관된 메모리 블록에, 메모리 블록(118)에 저장될 수 있거나 및/또는 다수의 블록들에 걸쳐 분산될 수 있다. CPU(102)에서 실행되는 명령들은 CPU(102)와 연관된 프로그램 메모리로부터 로딩될 수도 있거나 메모리 블록(118)으로부터 로딩될 수도 있다.

[0061] SOC (100) 는 또한, GPU (104), DSP (106), 5세대 (5G) 접속성, 4세대 롱 텀 에볼루션 (4G LTE) 접속성, Wi-Fi 접속성, USB 접속성, 블루투스 접속성 등을 포함할 수도 있는 접속성 블록 (110), 및 예를 들어, 제스처들을 검출 및 인식할 수도 있는 멀티미디어 프로세서 (112) 와 같은, 특정 기능들에 맞추어진 부가 프로세싱 블록들을 포함할 수도 있다. 일 구현에서, NPU 는 CPU (102), DSP (106), 및/또는 GPU (104) 에서 구현된다. SOC(100)는 또한 센서 프로세서(114), 이미지 신호 프로세서들(ISP들)(116), 및/또는 글로벌 포지셔닝 시스템을 포함할 수도 있는 내비게이션 모듈(120)을 포함할 수도 있다.

[0062] SOC (100) 는 ARM 명령 세트에 기초할 수도 있다. 본 개시의 일 양태에서, CPU(102)에 로딩된 명령들은 입력 값과 필터 가중치의 곱셈 곱에 대응하는 룩업 테이블(LUT)에서 저장된 곱셈 결과를 검색하기 위한 코드를 포함할 수 있다. CPU (102) 에 로딩된 명령들은 또한 곱셈 곱의 룩업 테이블 히트가 검출될 때 곱셈 곱의 곱셈 연산 동안 곱셈기를 디스에이블하기 위한 코드를 포함할 수도 있다. 또한, CPU (102) 에 로딩된 명령들은 곱셈 곱의 룩업 테이블 미스가 검출될 때 입력 값과 필터 가중치의 연산된 곱셈 곱을 저장하기 위한 코드를 포함할 수도 있다.

[0063] SOC (100) 및/또는 그 컴포넌트들은 본 명세서에서 논의된 본 개시의 양태들에 따른 머신 학습 기법들을 사용하여 비디오 압축 및/또는 압축해제 (또한 비디오 인코딩 및/또는 디코딩으로서 지칭되며, 집합적으로 비디오 코딩으로서 지칭됨) 를 수행하도록 구성될 수도 있다. 비디오 압축 및/또는 압축해제를 수행하기 위해 딥 러닝 아키텍처들을 사용함으로써, 본 개시의 양태들은 디바이스 상의 비디오 압축 및/또는 압축해제의 효율을 증가시킬 수 있다. 예를 들어, 설명된 비디오 코딩 기술을 사용하는 디바이스는 머신 학습 기반 기술을 사용하여 비디오를 보다 효율적으로 압축할 수 있고, 압축된 비디오를 다른 디바이스로 송신할 수 있으며, 다른 디바이스는 본 명세서에 설명된 머신 학습 기반 기술을 사용하여 압축된 비디오를 보다 효율적으로 압축 해제할 수 있다.

[0064] 전술한 바와 같이, 신경망은 머신 학습 시스템의 예이고, 입력 레이어, 하나 이상의 히든 레이어들, 및 출력 레이어를 포함할 수 있다. 입력 레이어의 입력 노드들에서 데이터가 제공되고 하나 이상의 히든 레이어들 중 히든 노드에 의해 프로세싱이 수행되며 출력 레이어의 출력 노드를 통해 출력이 생성된다. 딥 러닝 네트워크는 일반적으로 여러 히든 레이어들을 포함한다. 신경망의 각각의 레이어는 인공 뉴런들(또는 노드들)을 포함할 수 있는 특징 맵들 또는 활성화 맵들을 포함할 수 있다. 특징 맵은 필터, 커널 등을 포함할 수 있다. 노드들은 레이어들 중 하나 이상의 레이어들의 노드들의 중요도를 표시하는 데 사용되는 하나 이상의 가중치들을 포함할 수 있다. 일부 경우들에서, 딥 러닝 네트워크는 일련의 많은 히든 레이어들을 가질 수 있으며, 초기 레이어들은 입력의 단순하고 낮은 레벨 특성들을 결정하는 데 사용되고, 나중의 레이어들은 더 복잡하고 추상적인 특성들의 계층을 구축한다.

[0065] 딥 러닝 아키텍처는 특징들의 계층을 학습할 수도 있다. 예를 들어, 시각적 데이터가 제시되면, 제 1 레이어는 입력 스트림에서 에지들과 같은 비교적 단순한 특징들을 인식하는 것을 학습할 수도 있다. 또 다른 예에서, 청각 데이터가 제시되면, 제 1 레이어는 특정 주파수에서 스펙트럼 파워를 인식하는 것을 학습할 수도 있다. 제 1 레이어의 출력을 입력으로서 취하는 제 2 레이어는 시각적 데이터에 대한 단순한 형상 또는 청각적 데이터에 대한 소리의 조합과 같은 특징들의 조합을 인식하는 것을 학습할 수도 있다. 가령, 상위 레이어들은 시각 데이터에서의 복잡한 형상들 또는 청각 데이터에서의 단어들을 나타내는 것을 학습할 수도 있다. 더욱 상위 레이어들은 일반적인 시각적 객체 또는 구어를 인식하는 것을 학습할 수도 있다.

[0066] 딥 러닝 아키텍처들은 자연스러운 계층적 구조를 갖는 문제들에 적용될 때 특히 잘 수행할 수도 있다. 예를 들어, 자동차의 분류는 바퀴, 방풍 유리 및 다른 특징들을 인식하는 것을 먼저 학습하는 것으로부터 이익을 얻을

수도 있다. 이러한 특징들은 승용차, 트럭 및 비행기를 인식하기 위해 상이한 방식으로 상위 레이어들에서 결합될 수도 있다.

[0067] 신경망들은 다양한 연결성 패턴들로 설계될 수도 있다. 피드 포워드 망에서, 정보는 하위에서 상위 레이어들로 보내지며, 주어진 레이어에 있는 각 뉴런은 상위 레이어들에 있는 뉴런들과 통신한다. 계층적 표현은 전술한 바와 같이 피드 포워드 망의 연속적인 레이어들에 구축될 수도 있다. 신경망은 또한, 순환성 (recurrent) 또는 피드백 (feedback) (하향식이라고도 함) 연결들을 가질 수도 있다. 순환성 연결에서, 주어진 레이어에 있는 뉴런으로부터의 출력은 동일한 레이어에 있는 다른 뉴런에 전달될 수도 있다. 순환성 아키텍처는 시퀀스에서 신경망에 전달되는 입력 데이터 체크 중 하나를 초과하여 걸쳐있는 패턴들을 인식하는 데 도움이 될 수도 있다. 주어진 레이어에 있는 뉴런으로부터 하위 레이어에 있는 뉴런으로의 연결을 피드백 (또는 하향식) 연결이라고 한다. 많은 피드백 연결들을 갖는 네트워크는 상위 수준(high-level) 개념의 인식이 입력의 특정 하위 수준(low-level) 특징들을 구별하는 것을 보조할 수도 있을 때 도움이 될 수도 있다.

[0068] 신경망의 레이어들 사이의 연결들은 전체적으로 연결되거나 국부적으로 연결될 수도 있다. 도 2a 는 전체적으로 연결된 신경망(202)의 예를 예시한다. 전체 연결 신경망(202)에서, 제 1 레이어에서의 뉴런은 그의 출력을 제 2 레이어에서의 모든 뉴런에 전달할 수도 있어서, 제 2 레이어에서의 각각의 뉴런이 제 1 레이어에서의 모든 뉴런으로부터 입력을 수신할 것이다. 도 2b 는 국부적으로 연결된 신경망(204)의 예를 예시한다. 국부 연결 신경망(204)에서, 제 1 레이어에서의 뉴런은 제 2 레이어에서의 제한된 수의 뉴런들에 연결될 수도 있다. 보다 일반적으로, 국부적으로 연결된 신경망 (204) 의 국부적으로 연결된 레이어는 레이어에서의 각각의 뉴런이 동일하거나 유사한 연결성 패턴을 가질 것이지만, 상이한 값들 (예를 들어, 210, 212, 214, 및 216) 을 가질 수도 있는 연결 강도들을 갖도록 구성될 수도 있다. 주어진 영역에 있는 상위 레이어 뉴런이, 망에 대한 전체 입력의 제한된 부분의 특성에 맞게 훈련을 통해 조정된 입력을 수신할 수도 있기 때문에, 국부적으로 연결된 연결 패턴은 상위 레이어에서 공간적으로 구분되는 수용 필드들을 발생시킬 수도 있다.

[0069] 전술한 바와 같이, 디지털 비디오 데이터는 대량의 데이터를 포함할 수 있으며, 이는 통신 네트워크들뿐만 아니라 비디오 데이터를 프로세싱하고 저장하는 디바이스들에 상당한 부담을 줄 수 있다. 예를 들어, 압축되지 않은 비디오 콘텐츠를 기록하는 것은 일반적으로 기록된 비디오 콘텐츠의 해상도가 증가함에 따라 크게 증가하는 큰 파일 사이즈를 초래한다. 하나의 예시적인 예에서, 1080p/24에서 기록된 채널 당 압축되지 않은 16-비트 비디오 (예를 들어, 초당 24개의 프레임들이 캡처되는, 폭이 1920 픽셀들 및 높이가 1080 픽셀들인 해상도) 는 프레임당 12.4 메가바이트, 또는 초당 297.6 메가바이트를 차지할 수도 있다. 초당 24개의 프레임으로 4K 해상도로 기록된 채널당 압축되지 않은 16비트 비디오는 프레임당 49.8메가바이트, 즉 초당 1195.2메가바이트를 차지할 수도 있다.

[0070] 네트워크 대역폭은 큰 비디오 파일이 문제가 될 수 있는 또 다른 제약이다. 예를 들어, 비디오 콘텐츠는 종종 무선 네트워크를 통해 (예: LTE, LTE-Advanced, NR(New Radio), WiFiTM, 블루투스TM, 또는 기타 무선 네트워크를 통해) 전달되고, 소비자 인터넷 트래픽의 대부분을 차지할 수 있다. 무선 네트워크들에서 이용가능한 대역폭의 양의 발전에도 불구하고, 이들 네트워크들에서 비디오 콘텐츠를 전달하는데 사용되는 대역폭의 양을 감소시키는 것이 여전히 바람직할 수도 있다.

[0071] 압축되지 않은 비디오 콘텐츠는 물리적 저장을 위한 상당한 메모리 및 송신을 위한 상당한 대역폭을 수반할 수 있는 큰 파일들을 초래할 수 있기 때문에, 비디오 코딩 기법들이 그러한 비디오 콘텐츠를 압축한 다음 압축해제하기 위해 이용될 수 있다.

[0072] 비디오 콘텐츠의 사이즈 - 그리고 따라서 비디오 콘텐츠를 저장하는데 수반되는 저장량 - 및 비디오 콘텐츠를 전달하는데 수반되는 대역폭의 양을 감소시키기 위해, 다양한 비디오 코딩 기법들이 특히 HEVC, AVC, MPEG, VVC 와 같은 특정 비디오 코딩 표준에 따라 수행될 수 있다. 비디오 코딩은 종종, 비디오 이미지들 또는 시퀀스들에 존재하는 리던던시(redundancy) 를 이용하는 인터-예측 (inter-prediction), 인트라-예측 (intra-prediction)과 같은 예측 방법들을 사용한다. 비디오 코딩 기법들의 공통 목적은 비디오 품질의 열화를 회피 또는 최소화하면서 더 낮은 비트 레이트를 사용하는 형태로 비디오 데이터를 압축하는 것이다. 비디오 서비스에 대한 수요가 증가하고 새로운 비디오 서비스가 이용가능함에 따라, 더 나은 코딩 효율, 성능 및 레이트 제어를 갖춘 코딩 기술이 필요하다.

[0073] 일반적으로, 인코딩 디바이스는 인코딩된 비디오 비트스트림을 생성하기 위해 비디오 코딩 표준에 따라 비디오 데이터를 인코딩한다. 일부 예들에서, 인코딩된 비디오 비트스트림(또는 "비디오 비트스트림" 또는 "비트스트림")은 일련의 하나 이상의 코딩된 비디오 시퀀스들이다. 인코딩 디바이스는 각각의 특징을 다수의의

슬라이스들로 파티셔닝함으로써 화상들의 코딩된 표현들을 생성할 수 있다. 슬라이스는 그 슬라이스에서의 정보가 동일한 화상 내의 다른 슬라이스들로부터의 데이터에 의존하지 않고 코딩되도록 다른 슬라이스들에 대해 독립적이다. 슬라이스는 독립 슬라이스 세그먼트를 포함하는 하나 이상의 슬라이스 세그먼트들, 및 존재하는 경우, 이전 슬라이스 세그먼트들에 의존하는 하나 이상의 종속 슬라이스 세그먼트들을 포함한다. HEVC 에서, 슬라이스들은 루마 샘플들 및 크로마 샘플들의 코딩 트리 블록들 (CTB들) 로 파티셔닝된다. 루마 샘플들의 CTB와 크로마 샘플들의 하나 이상의 CTB들이, 그 샘플들을 위한 선택스와 함께, 코딩 트리 유닛(coding tree unit, CTU)이라고 지칭된다. CTU 는 "트리 블록" 또는 "최대 코딩 유닛" (LCU) 으로 지칭될 수도 있다. CTU 는 HEVC 인코딩을 위한 기본 프로세싱 유닛이다. CTU 는, 다양한 사이즈의 다수의 코딩 유닛들 (CU들) 로 분할될 수 있다. CU 는 코딩 블록 (CB) 들이라고 지칭되는 루마 및 크로마 샘플 어레이들을 포함한다.

[0074] 루마 및 크로마 CB 들은 예측 블록 (PB) 들로 더 분할될 수 있다. PB 는 (이용 가능하거나 사용을 위해 가능해질 때) 인터 예측 또는 인트라 블록 커피 (IBC) 예측에 동일한 모션 파라미터들을 사용하는 루마 성분 또는 크로마 성분의 샘플들의 블록이다. 루마 PB 및 하나 이상의 크로마 PB들은, 연관된 선택스와 함께, 예측 유닛 (PU) 을 형성한다. 인터 예측을 위해, 모션 파라미터들의 세트 (예를 들어, 하나 이상의 모션 벡터들, 레퍼런스 인덱스들 등) 가 각각의 PU 에 대해 비트스트림으로 시그널링되고, 루마 PB 및 하나 이상의 크로마 PB 들의 인터 예측을 위해 사용된다. 모션 파라미터들은 모션 정보로도 지칭될 수 있다. CB 는 또한 하나 이상의 변환 블록 (Tb)들로 파티셔닝될 수 있다. Tb 는, 예측 잔차 신호를 코딩하기 위해 잔차 변환 (예컨대, 일부 경우들에서 동일한 2 차원 변환) 이 적용되는 컬러 컴포넌트의 샘플들의 정사각형 블록을 나타낸다. 변환 유닛 (TU) 은 루마 및 크로마 샘플들의 Tb 들 및 대응하는 선택스 엘리먼트들을 나타낸다. 변환 코딩은 아래에 보다 상세히 설명한다.

[0075] HEVC 표준에 따르면, 변환들은 TU들을 이용하여 수행될 수도 있다. TU들은 주어진 CU 내의 PU들의 사이즈에 기초하여 사이징될 수도 있다. TU들은 통상적으로 PU들과 동일한 사이즈가거나 또는 PU들보다 더 작을 수도 있다. 일부 예들에서, CU 에 대응하는 잔차 샘플들은 잔차 쿼드트리 (residual quad tree; RQT) 로 알려진, 쿼드트리 구조를 이용하여 더 작은 유닛들로 세분될 수도 있다. RQT 의 리프 노드들은 TU들에 대응할 수도 있다. TU들과 연관되는 픽셀 차이 값들은 변환 계수들을 생성하도록 변환될 수도 있다. 그 다음, 변환 계수들은 인코딩 디바이스에 의해 양자화될 수도 있다.

[0076] 일단 비디오 데이터의 화상들이 CU들로 파티셔닝되면, 인코딩 디바이스는 예측 모드를 사용하여 각각의 PU 를 예측한다. 그 후, 예측 유닛 또는 예측 블록은 잔차들 (후술됨) 을 얻기 위해 원래의 비디오 데이터로부터 감산된다. 각각의 CU 에 대해, 예측 모드는 선택스 데이터를 사용하여 비트스트림 내에서 시그널링될 수도 있다. 예측 모드는 인트라 예측 (또는 인트라-화상 예측) 또는 인터 예측 (또는 인터-화상 예측) 을 포함할 수도 있다. 인트라 예측은 화상 내에서 공간적으로 이웃하는 샘플 간의 상관을 활용한다. 예를 들어, 인트라 예측을 사용하여, 각각의 PU 는, 예를 들어, PU 에 대한 평균값을 구하기 위한 DC 예측, PU 에 대해 평면 표면을 피팅 (fitting) 하기 위한 평면 예측, 이웃하는 데이터로부터 외삽하기 위한 방향 예측, 또는 임의의 다른 적절한 유형의 예측을 사용하여 동일한 화상 내의 이웃하는 이미지 데이터로부터 예측된다. 인터 예측은 이미지 샘플들의 블록을 위한 모션 보상된 예측을 도출하기 위해 화상들 간의 시간적 상관을 사용한다. 예를 들어, 인터 예측을 사용하여, 각각의 PU 는 (출력 순서에서 현재 화상의 전 또는 후의) 하나 이상의 참조 화상들에서의 이미지 데이터로부터의 모션 보상 예측을 사용하여 예측된다. 인터 화상 또는 인트라 화상 예측을 사용하여 화상 영역을 코딩할지 여부의 결정은 예를 들어 CU 레벨에서 행해질 수도 있다.

[0077] 인트라 및/또는 인터 예측을 이용하여 예측을 수행한 후, 인코딩 디바이스는 변환 및 양자화를 수행할 수 있다. 예를 들어, 예측에 이어서, 인코딩 디바이스는 PU 에 대응하는 잔차 값들을 산출할 수도 있다. 잔차 값들은 코딩된(PU) 픽셀들의 현재 블록과 현재 블록을 예측하는데 사용된 예측 블록(예컨대, 현재 블록의 예측된 버전) 사이의 픽셀 차이 값들을 포함할 수도 있다. 예를 들어, 예측 블록을 생성한 (예컨대, 인터 예측 또는 인트라 예측을 발행한) 후, 인코딩 디바이스는 현재 블록으로부터 예측 유닛에 의해 생성된 예측 블록을 감산함으로써 잔차 블록을 생성할 수 있다. 잔차 블록은 현재 블록의 픽셀 값과 예측 블록의 픽셀 값 사이의 차이를 정량화하는 픽셀 차이 값들의 세트를 포함한다. 일부 예들에서, 잔차 블록은 2 차원 블록 포맷 (예를 들어, 픽셀 값들의 2 차원 매트릭스 또는 어레이) 으로 표현될 수도 있다. 이러한 예에서, 잔차 블록은 픽셀 값들의 2 차원 표현이다.

[0078] 예측이 수행된 후에 남을 수 있는 임의의 잔차 데이터는 이산 코사인 변환, 이산 사인 변환, 정수 변환, 웨이블릿 변환, 다른 적절한 변환 함수 또는 이들의 임의의 조합에 기초할 수도 있는 블록 변환을 사용하여 변환된다. 일부 경우에서, 하나 이상의 블록 변환들 (예를 들어, 사이즈 32 x 32, 16 x 16, 8 x 8, 4 x 4, 또는 다른 적

합한 사이즈) 이 각각의 CU 에서 잔차 데이터에 적용될 수도 있다. 일부 실시형태들에서, TU 가 인코딩 디바이스에 의해 구현되는 변환 및 양자화 프로세스들을 위해 사용될 수도 있다. 하나 이상의 PU 들을 갖는 주어진 CU 는 하나 이상의 TU 들을 또한 포함할 수도 있다. 아래에 더 상세히 기술되는 바와 같이, 잔차 값들은 블록 변환을 사용하여 변환 계수로 변환될 수 있고, 그 후 TU를 사용하여 양자화되고 스캔되어 엔트로피 코딩을 위한 직렬화된 변환 계수를 생성할 수도 있다.

[0079] 인코딩 디바이스는 변환 계수들의 양자화를 수행할 수 있다. 양자화는 변환 계수를 양자화하여 그 계수를 나타내는 데 사용되는 데이터의 양을 줄임으로써 추가의 압축을 제공한다. 예를 들어, 양자화는 그 계수들의 일부 또는 전부와 연관되는 비트 깊이를 감소시킬 수도 있다. 일 예에서, n 비트 값을 갖는 계수는 양자화 동안 m 비트 값으로 라운드-다운될 수도 있으며, 여기서 n 은 m 보다 크다.

[0080] 일단 양자화가 수행되면, 코딩된 비디오 비트스트림은 양자화된 변환 계수, 예측 정보 (예를 들어, 예측 모드, 모션 벡터, 블록 벡터 등), 파티셔닝 정보, 및 다른 선택스 데이터와 같은 임의의 다른 적절한 데이터를 포함한다. 그 다음, 코딩된 비디오 비트스트림의 상이한 엘리먼트들은 인코딩 디바이스에 의해 엔트로피 인코딩될 수도 있다. 일부 예들에서, 인코딩 디바이스는 양자화된 변환 계수들을 스캔하기 위한 미리정의된 스캔 순서를 이용하여, 엔트로피 인코딩될 수 있는 직렬화된 벡터를 생성할 수도 있다. 일부 예들에서, 인코딩 디바이스는 적응적 스캔을 수행할 수도 있다. 벡터 (예를 들어, 1차원 벡터) 를 형성하기 위해 양자화된 변환 계수들을 스캔한 이후, 인코딩 디바이스는 그 벡터를 엔트로피 인코딩할 수도 있다. 예를 들어, 인코딩 디바이스는 컨텍스트 적응적 가변 길이 코딩, 컨텍스트 적응적 이진 산술 코딩, 선택스 기반 컨텍스트 적응적 이진 산술 코딩, 확률 인터벌 파티셔닝 엔트로피 코딩, 또는 다른 적합한 엔트로피 인코딩 기법을 사용할 수도 있다.

[0081] 인코딩 디바이스는 인코딩된 비디오 비트스트림을 저장할 수 있거나 및/또는 인코딩된 비디오 비트스트림 데이터를 통신 링크를 통해 디코딩 디바이스를 포함할 수 있는 수신 디바이스로 전송할 수 있다. 디코딩 디바이스는, 인코딩된 비디오 데이터를 구성하는 하나 이상의 코딩된 비디오 시퀀스들의 엘리먼트들을 (예컨대, 엔트로피 디코더를 사용하여) 엔트로피 디코딩하고 추출함으로써 인코딩된 비디오 비트스트림 데이터를 디코딩할 수도 있다. 그 다음, 디코딩 디바이스는 인코딩된 비디오 비트스트림 데이터에 대해 리스케일 (rescale) 하고 역 변환을 수행할 수도 있다. 그 다음, 잔차 데이터가 디코딩 디바이스의 예측 스테이지로 전달된다. 그 후, 디코딩 디바이스는 인트라 예측, 인터 예측, IBC, 및/또는 다른 타입의 예측을 사용하여 픽셀들의 블록 (예를 들어, PU) 을 예측한다. 일부 예들에 있어서, 예측은 역변환의 출력 (잔차 데이터) 에 추가된다. 디코딩 디바이스는 디코딩된 비디오를 비디오 목적지 디바이스에 출력할 수도 있으며, 비디오 목적지 디바이스는 디코딩된 비디오 데이터를 콘텐츠의 소비자에게 디스플레이하기 위한 디스플레이 또는 다른 출력 디바이스를 포함할 수도 있다.

[0082] 다양한 비디오 코딩 표준들에 의해 정의된 비디오 코딩 시스템들 및 기법들 (예를 들어, 상기 설명된 HEVC 비디오 코딩 기법들) 은 원시 비디오 콘텐츠에서 정보의 많은 부분을 유지할 수 있을 수도 있고, 신호 프로세싱 및 정보 이론 개념들에 기초하여 선형적으로 정의될 수도 있다. 그러나, 일부 경우들에서, 머신 학습(ML) 기반 이미지 및/또는 비디오 시스템은 엔드-투-엔드 신경망 기반 이미지 및 비디오 코딩(E2E-NNVC) 시스템과 같은 비-ML 기반 이미지 및 비디오 코딩 시스템들에 비해 이점들을 제공할 수 있다. 전술한 바와 같이, 많은 E2E-NNVC 시스템들은 오토인코더 서브네트워크(인코더 서브네트워크) 및 엔트로피 코딩에 사용되는 양자화된 잠재들을 통해 확률적 모델을 학습하는 것을 담당하는 제 2 서브네트워크의 조합으로서 설계된다. 이러한 아키텍처는 변환 플러스 양자화 모듈(인코더 서브-네트워크)과 엔트로피 모델링 서브네트워크 모듈의 조합으로 볼 수 있다.

[0083] 도 3 은 E2E-NNVC 시스템(310)을 사용하여 이미지 및/또는 비디오 인코딩 및 디코딩을 수행하도록 구성된 디바이스(302)를 포함하는 시스템(300)을 도시한다. 디바이스 (302) 는 카메라 (307) 및 저장 매체(314)(예를 들어, 데이터 저장 디바이스) 에 커플링된다. 일부 구현들에서, 카메라 (307) 는 E2E-NNVC 시스템 (310) 에 의한 인코딩을 위해 프로세서 (304) 에 이미지 데이터 (308)(예를 들어, 비디오 데이터 스트림) 를 제공하도록 구성된다. 일부 구현들에서, 디바이스 (302) 는 다수의 카메라들 (예를 들어, 듀얼 카메라 시스템, 3개의 카메라들, 또는 다른 수의 카메라들) 에 커플링될 수 있거나 및/또는 이들을 포함할 수 있다. 일부 경우들에서, 디바이스 (302) 는 마이크 및/또는 다른 입력 디바이스 (예를 들어, 키보드, 마우스, 터치스크린 및/또는 터치패드와 같은 터치 입력 디바이스, 및/또는 다른 입력 디바이스) 에 커플링될 수 있다. 일부 예들에서, 카메라 (307), 저장 매체 (314), 마이크, 및/또는 다른 입력 디바이스는 디바이스 (302) 의 일부일 수 있다.

[0084] 디바이스 (302) 는 또한 하나 이상의 무선 네트워크들, 하나 이상의 유선 네트워크들, 또는 이들의 조합과 같은 송신 매체 (318) 를 통해 제 2 디바이스 (390)에 커플링된다. 예를 들어, 송신 매체 (318) 는 무선 네트워크,

유선 네트워크, 또는 유선 및 무선 네트워크들의 조합에 의해 제공된 채널을 포함할 수 있다. 송신 매체 (318)는 패킷 기반 네트워크, 예를 들어 로컬 영역 네트워크, 광역 네트워크, 또는 글로벌 네트워크, 예를 들어 인터넷의 부분을 형성할 수도 있다. 송신 매체 (318)는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스로부터 목적지 디바이스로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다. 무선 네트워크가 임의의 무선 인터페이스 또는 무선 인터페이스들의 조합을 포함할 수도 있고, 임의의 적합한 무선 네트워크(예를 들어, 인터넷 또는 다른 광역 네트워크, 패킷 기반 네트워크, WiFi™, RF (radio frequency), UWB, WiFi-Direct, 셀룰러, LTE(Long-Term Evolution), WiMax™ 등)를 포함할 수도 있다. 유선 네트워크는 임의의 유선 인터페이스 (예 : 파이버, 이더넷, 전력선 이더넷, 동축 케이블을 통한 이더넷, DSL (Digital Signal Line) 등)를 포함할 수도 있다. 유선 및/또는 무선 네트워크는 기지국, 라우터, 액세스 포인트, 브리지, 게이트웨이, 스위치 등과 같은 다양한 장비를 사용하여 구현될 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라 변조되고, 수신 디바이스로 송신될 수도 있다.

[0085] 디바이스(302)는 메모리(306), 제1 인터페이스("I/F 1")(312) 및 제2 인터페이스("I/F 2")(316)에 커플링된 하나 이상의 프로세서(본 명세서에서 "프로세서"로 지칭됨)(304)를 포함한다. 프로세서 (304)는 카메라 (307)로부터, 메모리 (306)로부터, 및/또는 저장 매체 (314)로부터 이미지 데이터 (308)를 수신하도록 구성된다. 프로세서 (304)는 제 1 인터페이스 (312)를 통해 (예를 들어, 메모리 버스를 통해) 저장 매체 (314)에 커플링되고, 제 2 인터페이스 (316)(예를 들어, 네트워크 인터페이스 디바이스, 무선 트랜시버 및 안테나, 하나 이상의 다른 네트워크 인터페이스 디바이스들, 또는 이들의 조합)를 통해 송신 매체 (318)에 커플링된다.

[0086] 프로세서(304)는 E2E-NNVC 시스템(310)을 포함한다. E2E-NNVC 시스템 (310)은 인코더 부분 (362) 및 디코더 부분 (366)을 포함한다. 일부 구현들에서, E2E-NNVC 시스템 (310)은 하나 이상의 오토-인코더들을 포함할 수 있다. 인코더 부분 (362)은 입력 데이터 (370)를 수신하고 입력 데이터 (370)를 프로세싱하여 입력 데이터 (370)에 적어도 부분적으로 기초하여 출력 데이터 (374)를 생성하도록 구성된다.

[0087] 일부 구현들에서, E2E-NNVC 시스템(310)의 인코더 부분(362)은 출력 데이터(374)를 생성하기 위해 입력 데이터 (370)의 손실 압축을 수행하도록 구성되어, 출력 데이터(374)는 입력 데이터(370)보다 적은 비트들을 갖는다. 인코더 부분 (362)은 임의의 이전 표현들 (예를 들어, 하나 이상의 이전에 재구성된 프레임들)에 기초하여 모션 보상을 사용하지 않고서 입력 데이터 (370)(예를 들어, 이미지들 또는 비디오 프레임들)를 압축하도록 훈련될 수 있다. 예를 들어, 인코더 부분 (362)은 비디오 프레임을 그 비디오 프레임으로부터만의 비디오 데이터를 사용하여, 그리고 이전에 재구성된 프레임들의 임의의 데이터를 사용하지 않고서, 압축할 수 있다. 인코더 부분 (362)에 의해 프로세싱된 비디오 프레임들은 본 명세서에서 인트라-예측된 프레임 (I-프레임들)으로서 지칭될 수 있다. 일부 예에서, I-프레임들은 전통적인 비디오 코딩 기법을 사용하여(예를 들어, HEVC, VVC, MPEG-4 또는 다른 비디오 코딩 표준에 따라) 생성될 수 있다. 그러한 예에서, 프로세서(304)는 HEVC 표준에 관해 위에서 설명된 것과 같은 블록 기반 인트라 예측을 수행하도록 구성된 비디오 코딩 디바이스(예를 들어, 인코딩 디바이스)를 포함하거나 이와 커플링될 수도 있다. 이러한 예들에서, E2E-NNVC 시스템 (310)은 프로세서 (304)로부터 배제될 수 있다.

[0088] 일부 구현들에서, E2E-NNVC 시스템(310)의 인코더 부분(362)은 이전 표현들(예를 들어, 하나 이상의 이전에 재구성된 프레임들)에 기초하여 모션 보상을 사용하여 입력 데이터(370)(예를 들어, 비디오 프레임들)를 압축하도록 훈련될 수 있다. 예를 들어, 인코더 부분(362)은 그 비디오 프레임으로부터의 비디오 데이터를 사용하여 그리고 이전에 재구성된 프레임들의 데이터를 사용하여 비디오 프레임을 압축할 수 있다. 인코더 부분 (362)에 의해 프로세싱된 비디오 프레임들은 본 명세서에서 인트라-예측된 프레임 (P-프레임들)으로서 지칭될 수 있다. 모션 보상은 이전에 재구성된 프레임으로부터의 픽셀들이 잔차 정보와 함께 현재 프레임에서의 새로운 포지션들로 이동하는 방법을 기술하는 것에 의해 현재 프레임의 데이터를 결정하는데 사용될 수 있다.

[0089] 도시된 바와 같이, E2E-NNVC 시스템(310)의 인코더 부분(362)은 신경망(363) 및 양자화기(364)를 포함할 수 있다. 신경망(363)은, 하나 이상의 트랜스포머들, 하나 이상의 컨볼루션 신경망(CNN), 하나 이상의 완전-연결 신경망, 하나 이상의 게이트 순환 유닛(GRU), 하나 이상의 LSTM(Long Short-Term Memory) 네트워크, 하나 이상의 ConvRNN, 하나 이상의 ConvGRU, 하나 이상의 ConvLSTM, 하나 이상의 GAN, 이들의 임의의 조합, 및/또는 중간 데이터(372)를 생성하는 다른 유형의 신경망 아키텍처를 포함할 수 있다. 중간 데이터 (372)는 양자화기 (364)에 입력된다. 인코더 부분(362)에 포함될 수도 있는 컴포넌트들의 예들이 도 5a 및 도 6a에 예시되어 있다.

[0090] 양자화기(364)는 출력 데이터(374)를 생성하기 위해 중간 데이터(372)의 양자화 및 일부 경우들에서 엔트로피

코딩을 수행하도록 구성된다. 출력 데이터 (374) 는 양자화된 (및 일부 경우에 엔트로피 코딩된) 데이터를 포함할 수 있다. 양자화기 (364) 에 의해 수행되는 양자화 동작들은 중간 데이터 (372) 로부터 양자화된 코드들 (또는 E2E-NNVC 시스템 (310) 에 의해 생성된 양자화된 코드들을 나타내는 데이터) 의 생성을 야기할 수 있다. 양자화 코드(또는 양자화된 코드를 나타내는 데이터)는 또한 잠재 코드(latent code) 또는 잠재(latent) (z 로 표시됨) 로 지칭될 수 있다. 잠재에 적용되는 엔트로피 모델은 본 명세서에서 "프라이어"(prior)로 지칭될 수 있다. 일부 예에서, 양자화 및/또는 엔트로피 코딩 동작은 기존의 비디오 코딩 표준에 따라 비디오 데이터를 인코딩 및/또는 디코딩할 때 수행되는 기존의 양자화 및 엔트로피 코딩 동작을 사용하여 수행될 수 있다. 일부 예들에서, 양자화 및/또는 엔트로피 코딩 동작들은 E2E-NNVC 시스템 (310) 에 의해 행해질 수 있다. 하나의 예시적인 예에서, E2E-NNVC 시스템 (310) 은 지도 훈련을 사용하여 훈련될 수 있으며, 잔차 데이터는 입력으로서 사용되고, 양자화된 코드들 및 엔트로피 코드들은 훈련 동안 알려진 출력 (라벨들) 으로서 사용된다.

[0091] E2E-NNVC 시스템(310)의 디코더 부분(366)은 (예를 들어, 직접 양자화기(364)로부터 및/또는 저장 매체(314)로부터) 출력 데이터(374)를 수신하도록 구성된다. 디코더 부분 (366)은 출력 데이터 (374) 를 프로세싱하여 출력 데이터 (374) 에 적어도 부분적으로 기초하여 입력 데이터 (370) 의 표현 (376) 을 생성할 수 있다. 일부 예들에서, E2E-NNVC 시스템(310)의 디코더 부분(366)은 하나 이상의 트랜스포머들, 하나 이상의 CNN들, 하나 이상의 완전 연결 신경망들, 하나 이상의 GRU들, 하나 이상의 LSTM 네트워크들, 하나 이상의 ConvRNN들, 하나 이상의 ConvGRU들, 하나 이상의 ConvLSTM들, 하나 이상의 GAN들, 이들의 임의의 조합, 및/또는 다른 유형의 신경망 아키텍처들을 포함할 수도 있는 신경망(368)을 포함한다. 디코더 부분(366)에 포함될 수도 있는 컴포넌트들의 예들이 도 5b 및 도 6a에 예시되어 있다.

[0092] 프로세서 (304) 는 출력 데이터 (374) 를 송신 매체 (318) 또는 저장 매체 (314) 중 적어도 하나에 전송하도록 구성된다. 예를 들어, 출력 데이터 (374) 는 재구성된 데이터로서 입력 데이터 (370) 의 표현 (376) 을 생성하기 위해 디코더 부분 (366) 에 의한 나중의 취출 및 디코딩 (또는 압축 해제) 을 위해 저장 매체 (314) 에 저장될 수 있다. 재구성된 데이터는 출력 데이터 (374) 를 생성하기 위해 인코딩/압축된 비디오 데이터의 재생을 위한 것과 같은 다양한 목적을 위해 사용될 수 있다. 일부 구현들에서, 출력 데이터 (374) 는 재구성된 데이터로서 입력 데이터 (370) 의 표현 (376) 을 생성하기 위해 디코더 부분 (366) 에 매치하는 다른 디코더 디바이스에서 (예를 들어, 디바이스 (302)에서, 제 2 디바이스 (390) 에서, 또는 다른 디바이스에서) 디코딩될 수도 있다. 예를 들어, 제 2 디바이스(390)는 디코더 부분(366)과 매치하는(또는 실질적으로 매치하는) 디코더를 포함할 수도 있고, 출력 데이터(374)는 송신 매체(318)를 통해 제 2 디바이스(390)로 송신될 수도 있다. 제 2 디바이스 (390)는 입력 데이터(370)의 표현(376)을 재구성된 데이터로서 생성하기 위해 출력 데이터(374)를 프로세싱할 수 있다.

[0093] 시스템 (300) 의 컴포넌트들은, 하나 이상의 프로그래밍가능 전자 회로들 (예컨대, 마이크로프로세서들, 그래픽 프로세싱 유닛들 (GPU들), 디지털 신호 프로세서들 (DSP들), 중앙 프로세싱 유닛들 (CPU들), 및/또는 다른 적합한 전자 회로들) 을 포함할 수 있는 전자 회로들 또는 다른 전자적 하드웨어를 포함할 수 있거나 및/또는 이들을 이용하여 구현될 수 있거나, 및/또는 본원에 기술된 다양한 동작들을 수행하기 위해 컴퓨터 소프트웨어, 펌웨어, 또는 이들의 임의의 조합을 포함할 수 있거나 및/또는 이들을 이용하여 구현될 수 있다.

[0094] 시스템 (300) 이 특정 컴포넌트들을 포함하는 것으로 도시되지만, 당업자는 시스템 (300) 이 도 3 에 도시된 컴포넌트들보다 더 많거나 더 적은 컴포넌트들을 포함할 수 있다는 것을 이해할 것이다. 예를 들어, 시스템 (300) 은 또한 입력 디바이스 및 출력 디바이스 (도시되지 않음) 를 포함하거나 이를 포함하는 컴퓨팅 디바이스의 일부일 수 있다. 일부 구현들에서, 시스템(300)은 또한, 하나 이상의 메모리 디바이스들(예를 들어, 하나 이상의 랜덤 액세스 메모리(RAM) 컴포넌트들, 판독-전용 메모리(ROM) 컴포넌트들, 캐시 메모리 컴포넌트들, 버퍼 컴포넌트들, 데이터베이스 컴포넌트들, 및/또는 다른 메모리 디바이스들), 하나 이상의 메모리 디바이스들과 통신하거나 및/또는 그에 전기적으로 연결되는 하나 이상의 프로세싱 디바이스들(예를 들어, 하나 이상의 CPU들, GPU들, 및/또는 다른 프로세싱 디바이스들), 무선 통신을 수행하기 위한 하나 이상의 무선 인터페이스들(예를 들어, 각각의 무선 인터페이스에 대한 기저대역 프로세서 및 하나 이상의 트랜시버들을 포함함), 하나 이상의 하드와이어드 연결들을 통해 통신을 수행하기 위한 하나 이상의 유선 인터페이스들(예를 들어, 직렬 인터페이스, 이더넷 범용 직렬 버스(USB) 입력, 라이트닝 커넥터, 및/또는 다른 유선 인터페이스), 및/또는 도 3 에 도시되지 않은 다른 컴포넌트들을 포함할 수도 있거나 또는 그것들을 포함하는 컴퓨팅 디바이스의 일부일 수 있다.

[0095] 일부 구현들에서, 시스템(300)은 컴퓨팅 디바이스에 의해 국부적으로 구현되거나 및/또는 컴퓨팅 디바이스에 포함될 수 있다. 예를 들어, 컴퓨팅 디바이스는 모바일 디바이스, 개인용 컴퓨터, 태블릿 컴퓨터, 가상 현실 (VR)

디바이스 (예를 들어, 헤드 마운트 디스플레이 (HMD) 또는 다른 VR 디바이스), 증강 현실 (AR) 디바이스 (예를 들어, HMD, AR 안경, 또는 다른 AR 디바이스), 웨어러블 디바이스, 서버 (예를 들어, SaaS (software as a service) 시스템 또는 다른 서버 기반 시스템), 텔레비전, 및/또는 본 명세서에 설명된 기법들을 수행하기 위한 리소스 능력들을 갖는 임의의 다른 컴퓨팅 디바이스를 포함할 수 있다.

[0096] 일 예에서, E2E-NNVC 시스템(310)은 프로세서(304)에 커플링되고 프로세서(304)에 의해 실행가능한 명령들을 저장하도록 구성된 메모리(306), 및 안테나 및 프로세서(304)에 커플링되고 출력 데이터(374)를 원격 디바이스에 송신하도록 동작가능한 무선 트랜시버를 포함하는 휴대용 전자 디바이스에 통합될 수 있다.

[0097] 도 4는 이미지 및/또는 비디오 코딩을 위한 하이퍼프라이어 모델(hyperprior model)을 구현하기 위해 컨벌루셔널 신경망 레이어들을 사용하는 E2E-NNVC 시스템의 예를 제시하는 도면이다. 도 4의 E2E-NNVC 시스템에서의 g_a 및 g_s 서브네트워크들은 인코더 서브네트워크 (예를 들어, 인코더 부분 (362)) 및 디코더 서브네트워크 (예를 들어, 디코더 부분 (366))에 각각 대응한다. 도 4의 g_a 및 g_s 서브네트워크들은 3-채널 RGB 입력을 위해 설계되고, 여기서 3개의 R, G, 및 B 입력 채널들 모두가 동일한 신경망 레이어들(컨벌루셔널 레이어들 및 일반화된 분할 정규화(GDN) 레이어들)을 통과하고 이에 의해 프로세싱된다. (도 4에 도시된 것과 같은) E2E-NNVC 시스템들은 RGB 데이터 (여기서, 상이한 R, G, 및 B 채널들의 통계적 특성들은 유사하다) 및/또는 YUV 데이터와 같은 유사한 통계적 특성을 갖는 입력 채널들을 타겟팅할 수 있다. 그러나, 이전에 논의한 것처럼, 이미지 코딩을 수행하도록 훈련된 CNN은 오토그래시브 프라이어들(이는 느린 디코딩 시간을 초래함)을 기반으로 하는 것과 같이, 인코딩 또는 디코딩 동작 중 일방의 낮은 레이턴시 성능을 여전히 달성할 수 없다.

[0098] 이전에 언급한 바와 같이, 하나 이상의 트랜스포머 신경망들을 사용하여 이미지 및/또는 비디오 코딩(예를 들어, 낮은 레이턴시 인코딩 및 디코딩)을 수행하기 위한 시스템 및 기법이 본 명세서에서 설명된다. 트랜스포머 신경망은 예를 들어, 아래에 설명되는 도 4의 하이퍼프라이어 아키텍처 및/또는 도 6b의 SSF(scale-space flow) 아키텍처에 따라 조직된 트랜스포머 블록들 및/또는 트랜스포머 레이어들을 포함할 수 있다. 예를 들어, 도 4에 도시된 4개의 컨벌루셔널 네트워크 g_a , g_s , h_a , 및 h_s 는 대신에 대응하는 4개의 트랜스포머 신경망들로서 제공될 수 있으며, 이에 대해서는 아래에서 더 깊이 있게 설명될 것이다.

[0099] 일부 예들에서, 본 명세서에 설명된 하나 이상의 트랜스포머 기반 신경망들은 레이트 왜곡에 적어도 부분적으로 기초한 손실 함수를 사용하여 훈련될 수 있다. 왜곡은 원본 이미지(예: 인코더 서브네트워크에 입력으로서 제공되는 이미지)와 압축 해제/디코딩된 이미지(예: 디코더 서브네트워크에 의해 재구성되는 이미지) 사이의 평균 제곱 오차(MSE)로 결정될 수도 있다. 일부 예에서, 트랜스포머 기반 미디어 코딩 신경망을 훈련하는 데 사용되는 손실 함수는 라그랑주 승수를 사용하여 왜곡과 레이트 사이의 트레이드 오프에 기초할 수 있다. 이러한 레이트-왜곡 손실 함수의 한 예는

$$L = D + \beta * R$$
 이고, 여기서 D는 왜곡을 나타내고, R은 레이트를 나타내며, 상이한 β 값들은 상이한 비트레이트 및/또는 PSNR(peak-signal-to-noise ratio)에 대해 훈련된 모델들을 나타낸다.

[0100] 하나의 예시적인 예에서, 비지도 학습 기법(지도 없음)은 여기에 설명된 트랜스포머 기반 신경망들 중 하나 이상을 훈련하는 데 사용될 수 있다. 비지도 훈련 프로세스에서는, 훈련 데이터의 일부나 엘리먼트에 라벨링하거나 분류할 필요가 없을 수도 있다. 예를 들어, 역전파 훈련 프로세스를 사용하여 신경망(예를 들어, 도 5a 및 도 5b에 각각 도시된 것들과 같은 인코더 및/또는 디코더 서브네트워크)의 노드들의 가중치(및 일부 경우에는 바이어스와 같은 다른 파라미터)를 조정할 수 있다. 역전파는 순방향 패스, 손실 함수, 역방향 패스, 및 가중치 업데이트를 포함한다. 일부 예에서, 손실 함수는 위에서 설명된 레이트 왜곡 기반 손실 함수를 포함할 수 있다. 순방향 패스, 손실 함수, 역방향 패스, 및 파라미터 업데이트는 하나의 훈련 반복 동안 수행될 수 있다. 그 프로세스는 인코더 또는 디코더 서브네트워크의 파라미터들의 가중치들이 정확하게 튜닝될 때까지 훈련 데이터의 각각의 세트에 대해 특정 횟수의 반복들 동안 반복된다.

[0101] 실제 출력 값들이 훈련 데이터 출력과는 현저히 상이할 수도 있으므로, 손실 (또는 오차)는 제 1 훈련 데이터

입력들에 대해 높을 수도 있다. 훈련의 목표는 손실량(예: 손실 함수 $L = D + \beta * R$ 를 사용하는 것과 같이, 레이트 왜곡 손실)을 최소화하는 것이다. 신경망은 어느 입력들 (가중치들)이 신경망의 손실에 가장 기여했는지를 결정함으로써 역방향 패스를 수행하며, 가중치들을 조정하여 손실이 감소하고 결국 최소화된다. 가중치들에 대한 손실의 도함수(dL/dW 로 표시됨, 여기서 W는 특정 레이어에서의 가중치들이다)는 신경망의 손실에 가

장 기여한 가중치들을 결정하기 위해 계산된다. 예를 들어, 가중치들은 구매의 반대 방향으로 변화하도록 업데이트된다. 가중치 업데이트는

$$w = w_i - \eta \frac{dL}{dw}$$

로서 표기되며, 여기서 w 는 가중치를 표기하고, w_i 는 초기 가중치를 표기하며, η 는 학습 레이트를 표기한다. 학습 레이트는 임의의 적절한 값으로 설정되며, 높은 학습 레이트는 더 큰 가중치 업데이트들을 포함하고, 낮은 값은 더 작은 가중치 업데이트들을 표시한다. 인코더 또는 디코더 서브네트워크, 또는 하나의 그러한 서브네트워크의 구성 시프트된 윈도우 트랜스포머 레이어 또는 블록은 원하는 출력이 달성될 때까지 이러한 방식으로 계속 훈련된다. 일부 경우들에서, 인코더 서브네트워크 및/또는 디코더 서브네트워크의 각각의 컴포넌트는 유사한 방식으로 훈련된다.

[0102] 시프트된 윈도우 트랜스포머 블록들의 예시적인 아키텍처(도 6a), 시프트된 윈도우 트랜스포머 레이어들 사이의 패치 병합의 예(도 7a), 그리고 주어진 트랜스포머 레이어 내의 트랜스포머 블록들의 쌍에 의해 수행되는 시프트된 윈도우 셀프 어텐션의 예(도 7b)를 제공하는 도 6 내지 도 7b에 대한 논의로 넘어가기 전에, 아래에서 도 5a 및 도 5b를 간략하게 소개한다. 위의 예에 이어, 논의는 더 상세히 도 5a 및 도 5b로 돌아간다.

[0103] 도 5a는 일련의 트랜스포머 레이어들(520, 530, 540, 550)을 갖는 인코더 서브네트워크(500a)의 예시적인 아키텍처를 도시한다. 일부 예에서, 인코더 서브네트워크(500a)는 입력 이미지(502)를 잠재 표현(555a)으로 전환하는 비선형 변환을 수행하도록 훈련될 수 있다. 일부 경우들에서, 비선형 변환은 분석 변환이기도 하다. 예시된 바와 같이, 각각의 인코더 트랜스포머 레이어는 연속적으로 배열된 시프트된 윈도우 트랜스포머 블록들(524, 534, 544, 및 554)의 세트(본 명세서에서는 "인코더 트랜스포머 블록 세트들" 또는 "트랜스포머 블록 세트들"라고도 함)를 각각 포함할 수 있다. 일부 예에서, 인코더 트랜스포머 블록 세트들(524-554)은, 도 6a에 예시된 시프트된 윈도우 트랜스포머 블록들의 예시적인 쌍과 같은, 시프트된 윈도우 트랜스포머 블록들의 하나 이상의 쌍들을 포함할 수 있다. 예를 들어, 인코더 블록 세트들(524-554)로부터 인코더 트랜스포머 블록 세트에 제공되는 시프트된 윈도우 트랜스포머 블록들의 총 수는 2의 배수일 수 있다(예를 들어, 도 5a의 "x2", "x6" 및 유사한 표기법으로 예시되는 바처럼, 트랜스포머 블록 세트(524) 내 2개의 시프트된 윈도우 트랜스포머 블록들, 트랜스포머 블록 세트(534) 내 2개의 시프트된 윈도우 트랜스포머 블록들, 트랜스포머 블록 세트(544) 내 6개의 시프트된 윈도우 트랜스포머 블록들, 그리고 트랜스포머 블록 세트(554) 내 2개의 시프트된 윈도우 트랜스포머 블록들). 2개의 시프트된 윈도우 변압기 블록들을 갖는 변압기 블록 세트의 예가 도 6a에 도시되어 있으며 아래에 설명된다. 일부 경우에, 시프트된 윈도우 트랜스포머 블록들의 쌍은 연속적인 시프트된 윈도우 트랜스포머 블록들에 대한 셀프 어텐션을 계산하기 위해 2개의 파티셔닝 구성들 사이에 사용을 교번하는 2-단계 셀프 어텐션 산출 프로세스에 대응할 수 있다(예를 들어, 이에 대해서는 도 6a 및 도 7b에 관하여 설명될 것이다).

[0104] 도 5b는 패치 파티셔닝 엔진(510b) 및 일련의 트랜스포머 레이어들(560, 570, 580, 590)을 갖는 디코더 서브네트워크(500b)의 예시적인 아키텍처를 도시한다. 패치 파티셔닝 엔진(510b)은 인코더 서브네트워크(500a)의 패치 파티셔닝 엔진(510a)과 유사하게 동작할 수 있다. 일부 예에서, 디코더 서브네트워크(500b)는 잠재 표현(555b)을 재구성된 이미지(504)로 전환하는 비선형 변환을 수행하도록 훈련될 수 있다. 일부 경우들에서, 비선형 변환은 합성 변환이기도 하다. 잠재 표현(555b)은 인코더 서브네트워크(500a)로부터 디코더 서브네트워크(500b)로의 잠재 표현(555a)을 통신하는 양자화, 엔트로피 코딩 및/또는 데이터 송신 프로세스들에서 정보 손실이나 손상이 없는 경우에서와 같이 인코더 서브네트워크(500a)에 의해 출력되는 잠재 표현(555a)과 동일할 수 있다. 예시된 바와 같이, 각각의 디코더 트랜스포머 레이어는 연속적으로 배열된 시프트된 윈도우 트랜스포머 블록들(564, 574, 584, 및 594)의 세트(본 명세서에서는 "디코더 트랜스포머 블록 세트들" 또는 "트랜스포머 블록 세트들"라고도 함)를 각각 포함할 수 있다. 일부 예에서, 디코더 트랜스포머 블록 세트들(564-594)은, 도 6a에 예시된 시프트된 윈도우 트랜스포머 블록들의 예시적인 쌍과 같은, 시프트된 윈도우 트랜스포머 블록들의 하나 이상의 쌍들을 포함할 수 있다. 예를 들어, 트랜스포머 디코더 블록 세트들(564-594)로부터 디코더 트랜스포머 블록 세트에 제공되는 시프트된 윈도우 트랜스포머 블록들의 총 수는 2의 배수일 수 있다(예를 들어, 도 5b의 "x2", "x6" 및 유사한 표기법으로 예시되는 바처럼, 트랜스포머 블록 세트(564) 내 2개의 시프트된 윈도우 트랜스포머 블록들, 트랜스포머 블록 세트(574) 내 6개의 시프트된 윈도우 트랜스포머 블록들, 트랜스포머 블록 세트(584) 내 2개의 시프트된 윈도우 트랜스포머 블록들, 그리고 트랜스포머 블록 세트(594) 내 2개의 시프트된 윈도우 트랜스포머 블록들). 일부 경우에, 시프트된 윈도우 트랜스포머 블록들의 쌍은 연속적인 시프트된 윈도우 트랜스포머 블록들에 대한 셀프 어텐션을 계산하기 위해 2개의 파티셔닝 구성들 사이에 사용을 교번하는 2-단계 셀프 어텐션 산출 프로세스에 대응할 수 있다(예를 들어, 이는 도 6a 및 도 7b에 관한 것일 것이다).

[0105] 일부 예에서, 인코더 서브네트워크(500a)를 참조하여 본 명세서에서 이루어진 설명의 일부 또는 전부가 디코더 서브네트워크(500b)에도 적용될 수 있다는 점에 유의한다. 예를 들어, 일부 경우에서, 디코더 트랜스포머 레이어

들(560-590) 중 하나 이상은 인코더 트랜스포머 레이어들(520-550)의 시프트된 윈도우 트랜스포머 블록들과 동일하거나 그렇지 않으면 공통 아키텍처를 공유하는 시프트된 윈도우 트랜스포머 블록들을 활용할 수 있다. 더욱이, 일부 예에서 디코더 서브네트워크(500b)의 아키텍처는 인코더 서브네트워크(500a)의 아키텍처에 대칭일 수 있으며, 디코더 서브네트워크(500b)는 인코더 서브네트워크(500a)의 패치 병합 엔진(532, 542, 552) 대신 패치 병합 해제 엔진(563, 573, 583)을 사용한다. 디코더 서브네트워크(500b)와 인코더 서브네트워크(500a)의 아키텍처들 간의 대칭은 디코더 및 인코더 서브네트워크들에 제공된 구성 시프트된 윈도우 트랜스포머 블록들의 동일한 아키텍처 배열 또는 구성의 사용을 포함할 수 있다.

[0106] 도 5c 는 신경망 기반 이미지 및 비디오 코딩 시스템을 위한 트랜스포머 기반 엔드-투-엔드 신경망 아키텍처의 예를 예시하는 도면이다; 일부 경우에, 도 5c의 트랜스포머 기반 엔드 투 엔드 신경망 아키텍처는 도 5a의 인코더 서브네트워크(500a) 및 도 5b의 디코더 서브네트워크(500b)를 포함할 수 있다. 패치 분할 엔진(도 5c에서 "패치 분할"로 도시됨)은 도 5b에 도시된 패치 병합 해제 엔진과 유사하고 동일한 동작을 수행할 수 있다.

[0107] 이제, 도 5a 및 도 5b로 돌아가기 전에 도 6-7b에 묘사된 예들에 대해 논의한다. 이전에 언급된 바처럼, 도 6-7b는 시프트된 윈도우 트랜스포머 블록들의 예시적인 아키텍처(도 6a), 시프트된 윈도우 트랜스포머 레이어들 사이의 패치 병합의 예(도 7a), 그리고 주어진 트랜스포머 레이어 내의 트랜스포머 블록들의 쌍에 의해 수행되는 시프트된 윈도우 셀프 어텐션의 예를 제공한다.

[0108] 도 6a는, 트랜스포머 블록 쌍으로 총칭되는, 제 1 시프트된 윈도우 트랜스포머 블록(601)과 제 2 시프트된 윈도우 트랜스포머 블록(603)의 예시적인 아키텍처(600)를 묘사한다. 예시된 바와 같이, 제 1 시프트된 윈도우 트랜스포머 블록(601)은 레이어 노름(layer norm)(610a); 셀프 어텐션 컴포넌트(622)("제 1 셀프 어텐션 레이어"이라고도 함); 레이어 노름(612a); 및 다층 퍼셉트론(MLP)으로 묘사된, 피드포워드 신경망 컴포넌트(630a)를 포함한다. 제 2 시프트된 윈도우 트랜스포머 블록(603)은 레이어 노름(610b); 셀프 어텐션 컴포넌트(624)("제 2 셀프 어텐션 레이어"이라고도 함); 레이어 노름(612b); 및 다시 MLP로 묘사된, 피드포워드 신경망 컴포넌트(630b)를 포함한다.

[0109] 일부 예에서, 제 1 시프트된 윈도우 트랜스포머 블록(601)은, 서로 다른 윈도우 파티셔닝 구성들을 적용하는 그들 각각의 셀프 어텐션 레이어들(622 및 624)을 제외하고는, 제 2 시프트된 윈도우 트랜스포머 블록(603)과 동일할 수 있다. 일부 경우에, 제 1 셀프 어텐션 레이어(622)는 W-MSA(windowed multi-head self-attention)를 포함할 수 있고, 제 2 셀프 어텐션 레이어(624)는 SW-MSA(shifted window multi-head self-attention)를 포함할 수 있다.

[0110] 일부 예에서, 제 1 시프트된 윈도우 트랜스포머 블록(601)의 제1 셀프 어텐션 레이어(622)는 시프트 사이즈 = 0(시프트 없음에 대응함)인 시프트 윈도우를 사용할 수 있다. 예를 들어, 윈도우 사이즈가 8일 때, 제 1 어텐션 레이어(예: 제 1 셀프 어텐션 레이어(622))는 시프트 사이즈 = 0을 사용할 수 있고, 제 2 어텐션 레이어(제 2 셀프 어텐션 레이어(624)와 같은 제 1 어텐션 레이어를 뒤따름)은 시프트 사이즈 = 4를 사용할 수 있고, 제 3 어텐션 레이어(제 2 어텐션 레이어를 뒤따름)은 다시 시프트 사이즈 = 0을 사용할 수 있으며, 제 4 어텐션 레이어(제 3 어텐션 레이어를 뒤따름)은 시프트 사이즈 = 4를 사용할 수 있고, 주어진 트랜스포머 블록 세트에서 시프트된 윈도우 트랜스포머 블록들의 수에 대해 그렇게 계속된다(예를 들어, 도 5a 및 도 5b의 "x2", "x6" 및 유사한 표기법으로 예시됨). 시프트 사이즈 값들을 교번하는 것은(예: 시프트 = 0 및 시프트 = 4를 교번하는 것)은 윈도우들에 걸쳐 신호를 전파하는 효과를 가져온다. 일부 경우들에, 시프트 사이즈가 가변적일 수 있으므로, 시프트=0 다음에 시프트=4를 가질 필요가 없다.

[0111] 일부 예들에서, 제 1 셀프 어텐션 레이어(622)는 패치들의 세트를 다수의 패치들을 각각 포함하는 비중첩 윈도우들로 나누기 위해 비중첩 윈도우 파티셔닝 구성(이를테면 도 7b의 구성(720))을 적용할 수 있다. 그러면 제 1 셀프 어텐션 레이어(622)는 각각의 윈도우 내에서 국부적으로 셀프 어텐션을 계산할 수 있다. 제 1 셀프 어텐션 레이어(622)는 레이어 노름(612a) 레이어(예를 들어, Softmax 레이어)에 셀프 어텐션 정보를 제공할 수 있다. 예를 들어, 제 1 셀프 어텐션 레이어(622)는 다음과 같이 출력들의 매트릭스를 계산함으로써 셀프 어텐션 값들을 계산할 수 있다:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

[0112]

[0113] 매트릭스 $Q = W_q X$, 매트릭스 $K = W_k X$, 그리고 매트릭스 $V = W_v X$, 및 Q , K , V 에 대한 입력은

동일한 X (따라서 "셀프")이다. W_q , W_k , 및 W_v 항들은 입력 벡터 X 를 쿼리(Q), 키(K) 및 값(V) 매트릭스들에 투영하거나 맵핑하는 선형 레이어들이다. 항 d_k 는 키 k 의 차원을 나타내고, 여기서는 $\sqrt{d_k}$ 는 스케일링 팩터 (scaling factor)로 작용한다. Softmax는 셀프 어텐션 값들에 대한 가중치를 얻는 데 사용되는 Softmax 함수를 나타낸다. 레이어 노름(612a)은 가중치들을 피드포워드 신경망 컴포넌트(630a)(예를 들어, MLP(multi-layer perceptron) 레이어)로 출력할 수 있다. 그러면 제 1 시프트된 윈도우 트랜스포머 블록(601)의 출력은 제 2 시프트된 윈도우 트랜스포머 블록(603)에 대한 입력으로서 제공될 수 있다.

[0114] 제 2 셀프 어텐션 레이어(624)에서는, 윈도우 파티셔닝이 시프트되어, 그 결과 새로운 윈도우들이 제 1 셀프 어텐션 레이어(622)의 그것들과 중첩하게 된다. 예를 들어, 시프트된 윈도우 파티셔닝 구성(이를테면 도 7b의 구성(730))은 제 2 셀프 어텐션 레이어(624)에 의해 적용될 수 있다. 제 2 셀프 어텐션 레이어(624)의 시프트된 윈도우에서 셀프 어텐션 계산은 제 1 셀프 어텐션 레이어(622)에서의 이전 윈도우들의 경계들을 크로싱하여, 레이어 노름(612b)에 제공될 수 있는 크로스 윈도우 연결을 생성한다. 레이어 노름(612b)은 피드포워드 신경망 컴포넌트(630b)에 출력을 제공할 수 있다.

[0115] 셀프 어텐션을 국부적으로 결정함으로써, 본 명세서에 설명된 트랜스포머 기반 이미지 및 비디오 코딩 시스템은 낮은 레이턴시 인코딩 및 디코딩을 지원하는 더 큰 효율성과 계산 성능을 달성할 수 있다. 예를 들어, 일부 양태들에서, 트랜스포머 기반 이미지 및 비디오 코딩 시스템은 이미지 사이즈에 선형적인 계산적 복잡성을 갖는다. 제 2 셀프 어텐션 레이어를 통해 크로스 윈도우 연결을 획득함으로써, 트랜스포머 기반 이미지 및 비디오 코딩 시스템은 CNN 기반 및 다른 기존 접근법들과 연관된 레이트 왜곡 손실에 매치하거나 더 낮게하는 레이트 왜곡 손실을 달성할 수 있다.

[0116] 도 7a는 인코더 서브네트워크(500a)의 맥락에서 (예를 들어, 아래에서 위로) 더 깊은 인코더 트랜스포머 레이어들의 이미지 패치들을 병합하는 예시적인 프로세스를 예시하고, 이하에서 더 자세히 설명될 것이다. 도 7b는, 윈도우 파티셔닝 구성(720) 및 윈도우 파티셔닝 구성(730)을 포함하는, 2 개의 상이한 윈도우 파티셔닝 구성들의 예를 예시한다. 윈도우 파티셔닝 구성(720)은 패치 토큰들의 세트 상에 적용되는 비중첩 윈도우 파티셔닝을 묘사하며, 일부 예들에서, 도 6a의 제 1 셀프 어텐션 레이어(622)에 의해 활용될 수도 있다. 비중첩 윈도우 파티셔닝의 예는 722에서 표시되고, 그 구성 패치 토큰들 중 하나의 예는 711에서 표시된다. 일부 예들에서, 도 6a의 제 1 시프트된 윈도우 트랜스포머 블록(601)은 셀프-어텐션 컴포넌트(622)를 사용하여 비중첩 윈도우 파티셔닝 구성(720)을 적용할 수 있다.

[0117] 윈도우 파티셔닝 구성(730)은 패치 토큰들의 세트 상에 적용되는 시프트된 윈도우 파티셔닝을 묘사하며, 일부 예들에서, 도 6a의 제 2 셀프 어텐션 레이어(624)에 의해 활용될 수도 있다. 일부 경우들에서, 2 개의 윈도우 파티셔닝 구성들(720 및 730)은 패치 토큰들의 동일한 세트 상에 적용될 수 있다.

[0118] 일부 예에서, 비중첩 윈도우 파티셔닝 구성(720)은 입력 패치 토큰들의 세트를 동일한 사이즈의 윈도우들로 나누며, 여기서는 16개의 패치 토큰들을 포함하는 4×4 윈도우들로 도시되지만, 다른 윈도우 기하학적 구조 및/또는 사이즈들도 활용될 수 있다. 시프트된 윈도우 파티셔닝 구성(730)은 비중첩 파티셔닝 구성(720)의 윈도우들에 상대적으로 변위된 윈도우들을 활용할 수 있다. 예를 들어, 시프트된 윈도우들(732 및 734)은 파티셔닝 구성(720)의 비중첩 윈도우들 중 다수의 서로 다른 윈도우들에 이전에 포함되었던 토큰들의 세트를 각각 포함하도록 변위되었다. 단일 시프트된 윈도우에는 이전 셀프 어텐션 레이어의 다수의 비중첩 윈도우들로부터 패치 토큰들이 포함되어 있으므로, 이전에 언급한 크로스 윈도우 연결들이 그에 의해 도입될 수 있다. 예시된 바와 같이, 시프트된 윈도우 파티셔닝 구성(730)은 패치 토큰 세트의 경계들을 넘어 확장되는 윈도우 사이즈의 클리핑(clipping) 또는 절단(truncation)과 함께, 비중첩 윈도우 파티셔닝 구성과 동일한 4×4 윈도우 사이즈를 사용한다. 그러나, 일부 예들에서, 시프트된 윈도우 파티셔닝 구성(730) 및 비중첩 윈도우 파티셔닝 구성(720)은 서로 다른 윈도우 사이즈를 사용할 수 있다.

[0119] 이제 도 5a 및 5b에 도시된 인코더 및 디코더 서브네트워크들로 돌아가면, 도 5a에 묘사된 바처럼, 인코더 서브네트워크(500a)의 제 1 트랜스포머 레이어(520)는 2개의 시프트된 윈도우 트랜스포머 블록들(예를 들어, 시프트된 윈도우 트랜스포머 블록(524) 아래에 "x2" 라벨로 표시됨)을 포함하고; 트랜스포머 레이어(530)는 2개의 시프트된 윈도우 트랜스포머 블록들을 포함하고; 트랜스포머 레이어(540)는 6개의 시프트된 윈도우 트랜스포머 블록들을 포함하고; 트랜스포머 레이어(550)는 2개의 시프트된 윈도우 트랜스포머 블록들을 포함한다. 일부 예에서, 본 명세서에 설명된 트랜스포머 레이어들 중 하나 이상은 도 5a의 예에 도시된 것보다 더 많거나 더 적은

수의 시프트된 윈도우 트랜스포머 블록들을 각각 포함할 수 있다는 점에 유의한다.

[0120] 트랜스포머 레이어들(520-550)(및 그의 구성 시프트된 윈도우 트랜스포머 블록들)은 일부 예에서 비전 트랜스포머(vision transformer)를 기반으로 할 수 있다. 비전 트랜스포머는 이미지 기반 입력들에 대해 동작하도록 구성된 트랜스포머 아키텍처의 유형이다. 이미지 기반 입력들에는 정지 이미지들(예: 사진 또는 기타 유형의 정지 이미지들)과 비디오 이미지들(예: 비디오의 프레임들)이 포함될 수 있다. 비전 트랜스포머 아키텍처하에서, 입력 이미지는 먼저 복수의 비중첩 패치들로 파티셔닝된 다음, 이것들은 비전 트랜스포머(들)가 동작할 수도 있는 벡터를 얻기 위해 선형 공간에 투영된다.

[0121] 인코더 서브네트워크(500a)는 입력 이미지(502)를 비중첩 패치들로 분할하는 패치 파티셔닝 엔진(510a)을 포함한다(2 개의 패치 파티셔닝 동작들의 예가 도 7a에 묘사되어 있음). 입력 이미지(502)는 $H \times W \times 3$ 의 차원들을 갖는 것으로 표시되며, 여기서 H 는 높이(픽셀 단위)를 나타내고, W 는 너비(픽셀 단위)를 나타내며, 숫자 3은 입력 이미지의 차원성(입력 이미지의 각 색상 채널, 예를 들어 R, G 및 B에 대해 하나의 차원)을 나타낸다. 각각의 패치는, 예를 들어, 높이 및 너비 차원에서 픽셀 수로 주어지는, 동일하거나 유사한 사이즈를 가질 수 있다. 일부 예에서, 패치 파티셔닝 엔진(510a)에 의해 생성된 패치들의 사이즈는 미리 결정될 수 있다. 예시된 바와 같이, 미리 결정된 패치 사이즈는 4×4 이므로, 패치 파티셔닝 엔진(510a)에 의해 생성된 패치는 변당 4개의 픽셀이 있는 정사각형으로 배열된 16개의 픽셀을 포함하지만, 다른 사이즈 및/또는 높이 대 너비 비율도 활용될 수 있다. 추가적으로, 패치 파티셔닝 엔진(510)은 인코더 서브네트워크(500a)의 프로세스 흐름의 일부로서 입력 이미지(502)를 패치들로 파티셔닝할 수 있거나 또는 입력 이미지(502)를 미리 패치들로 파티셔닝할 수 있다는 점에 유의한다. 일부 예에서, 패치 파티셔닝 엔진(510a)에 의해 생성된 패치들은 토큰들("패치 토큰들"이라고도 함)으로 다루어지며, 여기서 각 토큰에 대한 특징은 패치의 구성 원시 픽셀 RGB 값들의 연쇄로서 설정된다. 예시된 예에서, 각각의 4×4 패치의 특징 차원은, 트랜스포머 레이어(520)에 제공된 입력 패치들의 세트를 자세히 설명하는 라벨 $\frac{H}{4} \times \frac{W}{4} \times 48$ 에 표시된 대로, 48 (예를 들어, $4 \times 4 \times 3 = 48$)이다.

[0122] 패치 파티셔닝 모델(510a)로부터, 입력 이미지(502)로부터 생성된 비중첩 패치들의 세트는 제 1 트랜스포머 레이어(520)에 제공된다. 예시된 바와 같이, 제 1 트랜스포머 레이어(520)는 선형 임베딩 레이어(521)를 포함하고, 이것은 각각의 패치의 원시 값의 특징들에 대해, 그것들을 임의의 차원 (C) 으로 투영시키기 위하여, 이를테면 선형 변환을 적용함으로써, 적용된다. 일부 예에서, 선형 임베딩 레이어(521)는 제 1 트랜스포머 레이어(520) 외부에 있거나 그렇지 않으면 그로부터 분리될 수 있고, 예를 들어 패치 파티셔닝 엔진(510a)과 조합하여 제공될 수도 있다. 또한, 선형 임베딩 레이어(521)는 제 1 트랜스포머 레이어(520)에만 나타난다 - 연속적인 트랜스포머 레이어들(530, 540, 550)에는 선형 임베딩 레이어가 결여되어 있지만 패치 병합 엔진들(532, 542, 552)을 각각 포함한다.

[0123] 다음으로 시프트된 윈도우 트랜스포머 블록들(524)이 패치 토큰들 상에 적용된다. 이전에 언급된 바와 같이, 일련의 시프트된 윈도우 트랜스포머 블록들(524)은 예를 들어 도 6a에 도시된 두 개와 같은 두 개의 연속적인 시프트된 윈도우 트랜스포머 블록들을 포함하는 것으로 도시되어 있다. 시프트된 윈도우 트랜스포머 블록들(524)은 수정된 셀프-어텐션 계산(도 6 및 도 7a-b를 참조하여 더 깊이 있게 설명됨)을 수행하여 하나 이상의 어텐션 벡터들을 생성한 다음, 이것들은 특징 변환에 사용된다. 시프트된 윈도우 트랜스포머 블록들(524)의 출력은 이어서 제 2 트랜스포머 레이어(530)에 대한 입력으로서 제공되며, 여기서 패치 토큰들의 수(예를 들어, $\frac{H}{4} \times \frac{W}{4}$) 및 선형 임베딩 양자 모두가 유지된다.

[0124] 입력 이미지(502)로부터 계층적 표현 또는 특징 맵을 생성하기 위해, 트랜스포머 레이어들(530, 540, 및 550) 각각은 각각의 패치 병합 엔진들(532, 542, 552)을 포함한다. 패치 병합 엔진(532, 542, 552)은 트랜스포머 레이어가 동작할 패치 토큰들의 수를 줄이기 위해 적용될 수 있다. 예시된 바와 같이, 패치 병합 엔진들(532-552)은 각 트랜스포머 레이어에서의 시프트된 윈도우 트랜스포머 블록들(534-554) 이전에 제공된다. 일부 예에서, 트랜스포머 레이어들(530-550) 중 하나 이상은 그의 패치 병합 엔진의 출력이 그의 일련의 시프트된 윈도우 트랜스포머 블록들의 입력에 직접 커플링되도록 구성될 수 있다.

[0125] 일부 예들에서, 패치 병합 엔진(532)은 선형 트랜스포머 레이어(520)로부터의 패치들의 세트를 이웃하는 패치들 또는 블록들의 비중첩 그룹들로 나누도록 구성된다. 패치 병합 엔진(532)은 각각의 그룹 내의 패치 특징들을 연쇄시킬 수 있고 각각의 연쇄된 그룹에 대해 단일의 새로운 패치를 생성할 수 있다. 하나의 예시적인 예에서, 패치 병합 엔진들(532, 542, 552)에 의해 수행되는 패치 병합 동작은 공간 데이터의 블록들을 깊이로 재배열하는

것을 포함할 수 있다. 특히, 패치 병합 동작은 높이 및 너비 차원들의 값들이 깊이 차원으로 이동되는 입력 텐서의 사본(copy)을 출력하는 것을 포함할 수 있다. $block_size \times block_size$ (여기서 $block_size$ 는 깊이/채널 차원으로 재배열될 입력 블록들 또는 패치의 공간 사이즈를 나타냄)의 크기의 비중첩 패치들 또는 블록들은 블록들 또는 패치의 각각의 위치에서 깊이로 재배열된다(왜냐하면 사이즈 $block_size \times block_size$ 의 각각의 입력 패치는 출력에서 1 픽셀로 붕괴(collapse)될 것이기 때문이다). 출력 텐서의 깊이는 $block_size * block_size * input_depth$ 이다. 입력의 각 블록 내의 수직(Y) 및 수평(X) 좌표들은 출력 채널 인덱스의 상위 컴포넌트가 된다. 입력 텐서의 높이와 너비는 $block_size$ 로 나누어질 수 있다. 일부 예들에서, 인코더 서브네트워크(500a)는 특징들의 채널 수를 변경하기 위해 패치 병합 동작들 후에 선형 레이어들을 적용할 수 있다. 패치 병합 엔진(542)과 패치 병합 엔진(552)에 의해 유사한 동작들이 수행될 수 있다. 따라서, 패치 병합 엔진들(532-552)은 병합된 그룹당 패치들의 수와 동일한 팩터만큼 각각의 연속적인 트랜스포머 레이어(530-550)에 제공되는 패치 토큰들의 수를 감소시킬 수 있다. 위에 언급된 바처럼, 도 7a는 인코더 서브네트워크(500a)의 맥락에서 (예를 들어, 아래에서 위로) 더 깊은 인코더 트랜스포머 레이어들의 이미지 패치들을 병합하는 예시적인 프로세스를 예시한다.

[0126] 도 5b를 참조하면, 디코더 서브네트워크(500b)의 트랜스포머 레이어들(560, 570, 및 580) 각각은 각각의 패치 병합 해제 엔진(563, 573, 및 583)(패치 분할 엔진으로도 지칭됨)을 포함한다. 패치 병합 해제 엔진들(563, 573, 583)은 특징 맵을 업샘플링(사이즈 증가)하기 위해 패치 병합 해제 프로세스를 수행할 수 있다. 패치 병합 해제 프로세스는 패치 병합 엔진들(532-552)에 의해 수행되는 패치 병합 프로세스의 역 프로세스이다. 일부 예에서, 패치 병합 해제 엔진(563)은 선행 블록 또는 레이어(예를 들어, 동일한 트랜스포머 레이어에서의 시프트된 윈도우 트랜스포머 블록)로부터의 이웃 패치들의 비중첩 그룹들을 4개 채널을 갖는 하나의 픽셀에서 하나의 채널이 있는 4개의 픽셀(업스케일 팩터가 2인 경우에) 분할하도록 구성된다. 예를 들어, 하나의 예시적인 예에서, 패치 병합 해제 엔진(563, 573, 및 583)에 의해 수행되는 패치 병합 해제 동작은 형상 $(* , C \times r^2 , H , W)$ 의 텐서에서 엘리먼트들을 형상 $(* , C , H \times r , W \times r)$ 의 텐서로 재배열하는 것을 포함할 수 있고 여기서 r 은 업스케일 팩터이다. 일부 예들에서, 디코더 서브네트워크(500b)는 특징들의 채널 수를 변경하기 위해 패치 병합 해제 동작들 후에 선형 레이어들을 적용할 수 있다. 도 7a는 또한 디코더 서브네트워크(500b)의 맥락에서 (예를 들어, 위에서 아래로) 더 깊은 디코더 트랜스포머 레이어들에서 이미지 패치들을 병합 해제하는 예시적인 프로세스를 예시하는 것으로 보여질 수 있다.

[0127] 제 1 레이어(702)에서, 이미지 데이터는 패치들의 16×16 그리드로 파티셔닝된 것으로 표현되며, 여기서 각각의 패치는 원본 입력 이미지(502)로부터 일부 수의 별개의 픽셀들을 포함한다. 패치 병합은 제 1 레이어(702)과 제 2 레이어(704) 사이에, 그리고 제 2 레이어(704)와 제 3 레이어(706) 사이에 적용된다. 일부 예에서, 레이어들(702, 704, 706)은 트랜스포머 레이어들(520, 530, 540)과 각각 동일할 수 있고, 위에 설명된 패치 병합 프로세스는 패치 병합 엔진들(532 및 542)에 의해 수행될 수 있다.

[0128] 제 1 레이어(702)와 제 2 레이어(704) 사이에서, 패치 병합은 (예를 들어, 제 1 레이어(702)의) 이웃한 패치들의 2×2 그룹들로부터의 특징들을 (예를 들어, 제 2 레이어(704)의) 단일 패치로 연쇄시키고 선형 레이어를 연쇄된 특징들에 적용함으로써 수행된다. 따라서, 제 2 레이어(704)에서의 단일 병합된 패치는 제 1 레이어(702)에서의 4개의 개별 패치들로부터 연쇄된 특징들을 포함한다. 패치들의 총 수는 4의 팩터만큼 줄어들었고 해상도는 2의 팩터만큼 다운샘플링되었으며 패치 병합 프로세스의 출력 차원은 $2C$ 이다.

[0129] 제 2 레이어(704)와 제 3 레이어(706) 사이에서, 그리고 동일한 패치 병합 프로세스가 적용될 수 있다(예를 들어, 여기서 2×2 이웃 패치들의 그룹들이 단일 패치로 병합됨). 제 3 레이어(706)에서의 결과적인 병합된 패치는 제 2 레이어(704)에서의 4개의 개별 패치들로부터 연쇄된 특징들을 포함하거나, 또는 대안적으로 제 1 레이어(702)에서 16개의 개별 패치들로부터 연쇄된 특징들을 포함하는 것으로 볼 수 있다. 제 1 레이어(702)와 제 2 레이어(704) 사이의 경우와 마찬가지로, 제 2 레이어(704)와 제 3 레이어(706) 사이의 패치 병합 프로세스는 다시 패치들의 총 수를 4의 팩터만큼 줄이고 해상도를 2의 팩터만큼 다운샘플링한다. 따라서, 이 패치 병합 프로세스의 출력 차원은 $4C$ 이다.

[0130] 따라서, 반복되는 패치 병합 로직이 활용되는 경우, 패치 병합 엔진들(532-552)은, 각각의 패치 병합 엔진이 서로 다른 입력 및 출력 해상도를 갖더라도 (예를 들어, 패치 병합 엔진(532)이 $\frac{H}{4} \times \frac{W}{4} \times C$ 의 입력 패치 토큰 해

상도 및 $\frac{H}{8} \times \frac{W}{8} \times 2C$ 의 출력 해상도를 가지며; 패치 병합 엔진(542)이 $\frac{H}{8} \times \frac{W}{8} \times 2C$ 의 입력 해상도와 $\frac{H}{16} \times \frac{W}{16} \times 4C$ 의 출력 해상도를 가지며; 그리고 패치 병합 엔진(552)이 $\frac{H}{16} \times \frac{W}{16} \times 4C$ 의 입력 해상도와 $\frac{H}{32} \times \frac{W}{32} \times 8C$ 의 출력 해상도를 갖는다) 서로 동일할 수 있다.

[0131] 도 6b는 본 명세서에서 설명된 하나 이상의 트랜스포머 기반 신경망 아키텍처를 활용할 수 있는 비디오 코딩 아키텍처의 예를 예시하는 도면이다. 트랜스포머 기반 신경망을 사용하면 비디오 코딩 아키텍처가 낮은 레이턴시(예: 낮은 지연 비디오 코딩 아키텍처)으로 동작할 수 있다. 도 6b의 비디오 코딩 시스템의 아키텍처는 시스템이 불확실성을 더 잘 모델링할 수 있도록 스케일 파라미터를 적어도 부분적으로 추가하여 광학 흐름을 확장하거나 일반화하는 스케일 공간 흐름(SSF)을 기반으로 한다. 도 6b의 맥락에서, 스케일 파라미터는 스케일 공간 위평 컴포넌트와 연관된다. 예시된 바와 같이, SSF 비디오 코딩 아키텍처는 모션 보상을 수행하기 위해, 집합적으로 "흐름 AE" 또는 흐름 자동 인코더로 라벨링된, SSF 인코더 및 SSF 디코더를 사용한다. SSF 인코더와 SSF 디코더는 둘 다 하나 이상의 컨벌루션 변환에 기초할 수도 있다. 유사하게, 잔차 인코더와 잔차 디코더는 또한 둘 다 하나 이상의 컨벌루션 변환에 기초할 수도 있다. 일부 경우들에서, SSF 인코더와 디코더 그리고 잔차 인코더와 디코더는 각각 CNN들로서 구현될 수도 있다. 일부 예들에서, 4개 모두의 인코더 및 디코더 컴포넌트들은 4-레이어 CNN들로서 제공될 수 있다.

[0132] 도 6b에 도시된 바와 같이, 본 명세서에 설명된 하나 이상의 시프트된 윈도우 트랜스포머 신경망들은 일부 예에서 SSF 아키텍처에서 CNN들 및/또는 컨벌루션 변환들을 대체하는 데 사용될 수 있다. 예를 들어, SSF 인코더, SSF 디코더, 잔차 인코더 및/또는 잔차 디코더를 대체하기 위해 하나 이상의 시프트된 윈도우 트랜스포머 신경망들이 제공될 수 있다. 일부 예에서, 4개의 시프트된 윈도우 트랜스포머 신경망들은, 아래에서 더 깊이 설명되는 바와 같이, SSF 비디오 코딩 아키텍처에서 4개의 컨벌루션 컴포넌트들을 대체할 수 있다.

[0133] 도 8는 이미지 및/또는 비디오 데이터를 프로세싱하기 위한 프로세스(800)의 예를 예시하는 플로우차트이다. 블록(802)에서, 프로세스(800)는 인코딩된 이미지 데이터의 프레임의 잠재 표현을 획득하는 것을 포함한다. 일부 예들에서, 인코딩된 이미지 데이터의 프레임은 인코딩된 정지 이미지를 포함한다. 일부 예들에서, 인코딩된 이미지 데이터의 프레임은 인코딩된 비디오 프레임을 포함한다.

[0134] 블록(804)에서, 프로세스(800)는, 입력으로서 인코딩된 이미지 데이터의 프레임의 잠재 표현을 사용하여 디코더 서브네트워크의 복수의 디코더 트랜스포머 레이어들에 의해, 디코딩된 이미지 데이터의 프레임을 생성하는 것을 포함한다. 일부 경우들에서, 복수의 디코더 트랜스포머 레이어들은 일련의 연속적인 디코더 트랜스포머 레이어들(예를 들어, 도 5b에 도시된 일련의 디코더 트랜스포머 레이어들(560, 570, 580, 및 590))을 포함한다. 복수의 디코더 트랜스포머 레이어들 중 적어도 하나의 디코더 트랜스포머 레이어는 특징들의 하나 이상의 패치들을 생성하고 하나 이상의 패치들 상에 적용된 하나 이상의 윈도우 파티션들 및 시프트된 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하기 위한 하나 이상의 트랜스포머 블록들을 포함한다. 복수의 디코더 트랜스포머 레이어들 중 적어도 하나의 디코더 트랜스포머 레이어는 하나 이상의 패치들 중 각각의 패치의 각각의 사이즈를 감소시키기 위한 패치 병합 해제 엔진을 더 포함한다. 예를 들어, 예시적인 예로서 도 5b를 참조하면, 디코더 트랜스포머 레이어(560)는 시프트된 윈도우 트랜스포머 블록(564) 및 패치 병합 해제 엔진(563)을 포함한다. 일부 양태들에서, 적어도 하나의 디코더 트랜스포머 레이어에 포함된 하나 이상의 트랜스포머 블록들 중 적어도 일부는 동일한 아키텍처를 갖는다. 일부 양태들에서, 적어도 하나의 디코더 트랜스포머 레이어에 포함된 하나 이상의 트랜스포머 블록들의 각각(예를 들어, 전부)은 동일한 아키텍처를 갖는다.

[0135] 일부 예들에서, 디코딩된 이미지 데이터의 프레임을 생성하기 위해, 프로세서(800)는 복수의 디코더 트랜스포머 레이어들 중 제 1 디코더 트랜스포머 레이어의 제 1 트랜스포머 블록에 의해, 하나 이상의 패치들 상에 적용된 하나 이상의 제 1 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것을 포함할 수 있다. 프로세스(800)는, 제 1 디코더 트랜스포머 레이어의 제 2 트랜스포머 블록에 의해, 하나 이상의 패치들 상에 적용된 하나 이상의 제 2 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것을 더 포함할 수 있다. 예를 들어, 하나 이상의 제 2 윈도우 파티션들은 (예를 들어, 도 7a 및 도 7b에 도시된 바와 같이) 하나 이상의 제 1 윈도우 파티션들 중 인접한 것들 사이의 하나 이상의 경계들과 중첩되도록 시프트될 수도 있다. 프로세스(800)는, 패치 병합 해제 엔진에 의해, 하나 이상의 패치들의 각각의 패치를 복수의 병합 해제된 패치들로 세그먼트화하는 것을 포함할 수 있다. 복수의 병합 해제된 패치들은 중첩되지 않는다. 일부 예들에서, 복수의 병합 해제

된 패치들 중 각각의 병합 해제된 패치는 균일한 패치 사이즈를 갖는다. 일부 경우들에서, 패치 병합 해제 엔진은 2 또는 다른 값의 패치 사이즈 감소 팩터를 적용한다. 일부 양태에서, 복수의 병합 해제된 패치들로 하나 이상의 패치들 중 각각의 패치를 세그먼트화하기 위해, 프로세스(800)는 복수의 병합 해제된 패치들의 특징 차원(feature dimension)을 감소시키는 것을 포함할 수 있다.

[0136] 일부 경우들에서, 프로세스(800)는 복수의 디코더 트랜스포머 레이어들 중 제 2 디코더 트랜스포머 레이어의 제 1 트랜스포머 블록에 복수의 병합 해제된 패치들을 제공하는 것을 포함할 수 있다. 예를 들어, 디코더 트랜스포머 레이어(560)의 패치 병합 해제 엔진(563)은 디코더 트랜스포머 레이어(570)의 트랜스포머 블록(574)에 병합 해제된 패치들을 제공할 수 있다. 일부 예에서, 프로세스(800)는, 제 2 디코더 트랜스포머 레이어의 패치 병합 해제 엔진에 의해, 복수의 병합 해제된 패치들을 세그먼트화하는 것을 포함할 수 있다. 프로세스(800)는 패치 병합 해제 엔진의 출력을 복수의 디코더 트랜스포머 레이어들 중 제 3 디코더 트랜스포머 레이어에 제공하는 것을 더 포함할 수 있다. 예를 들어, 디코더 트랜스포머 레이어(570)의 패치 병합 해제 엔진(573)은 디코더 트랜스포머 레이어(580)(예를 들어, 디코더 트랜스포머 레이어(580)의 트랜스포머 블록(584))에 병합 해제된 패치들을 제공할 수 있다. 다른 예에서, 디코더 트랜스포머 레이어(590)의 선형 임베딩 엔진(591)은 트랜스포머 블록(594)으로부터 출력된 특징들에 선형 변환을 적용할 수 있고 선형 변환된 특징들을 패치 파티셔닝 엔진(510b)으로 출력할 수 있다.

[0137] 일부 양태에서, 프로세스(800)는, 복수의 디코더 트랜스포머 레이어들에 의해, 입력으로서 인코딩된 이미지 데이터의 프레임의 잠재 표현을 수신하는 것을 포함할 수 있다. 프로세스(800)는 디코딩된 이미지 데이터의 프레임 임을 생성하기 위해 비선형 변환을 적용하는 것을 포함할 수 있다. 일부 경우들에서, 비선형 변환은 합성 변환이고, 디코딩된 이미지 데이터의 프레임은 인코딩된 이미지 데이터의 프레임과 연관된 입력 이미지의 재구성이다.

[0138] 일부 예들에서, 프로세스(800)는 레이트-왜곡(rate-distortion)에 적어도 부분적으로 기초한 손실 함수를 사용하여 복수의 디코더 트랜스포머 레이어들 중 하나 이상의 디코더 트랜스포머 레이어들을 훈련하는 것을 포함할 수 있다. 일부 경우들에서, 손실 함수는 레이트 왜곡을 위한 라그랑지 승수(Lagrangian multiplier)를 포함한다. 예를 들어, 본 명세서에서 설명된 바처럼, 레이트-왜곡 손실 함수는
$$L = D + \beta * R$$
를 포함할 수 있고, 여기서 D는 왜곡을 나타내고, R은 레이트를 나타내며, 상이한 β 값들은 상이한 비트레이트 및/또는 PSNR(peak-signal-to-noise ratio)에 대해 훈련된 모델을 나타낸다.

[0139] 일부 예들에서, 프로세스(800)는 적어도 제 1 훈련 데이터 세트 및 제 2 훈련 데이터 세트로 (예를 들어, 비지도 훈련 또는 학습을 사용하여) 복수의 디코더 트랜스포머 레이어들을 훈련하는 것을 포함할 수 있다. 그러한 예들에서, 제 2 훈련 데이터 세트의 데이터는 제 1 훈련 데이터 세트의 데이터와 비교하여 역 시간 순서를 갖는다. 일부 경우들에서, 트랜스포머 블록들 내의 모든 LayerNorm 레이어들을 제거할 수 있다. 일부 예에서는, 트랜스포머 블록들에 사용되는 윈도우 사이즈를 8에서 4로 줄일 수 있다. 제 1 훈련 세트의 데이터와 비교하여 제 2 훈련 세트의 데이터에 대해 역 시간 순서를 사용하면, 트랜스포머 블록들 내의 LayerNorm 레이어들을 제거하거나 및/또는 윈도우 사이즈를 8에서 4로 줄이는 것이 안정적인 훈련으로 이어질 수 있다.

[0140] 도 9는 이미지 및/또는 비디오 데이터를 프로세싱하기 위한 프로세스 (900)의 예를 예시하는 플로우차트이다. 블록(902)에서, 프로세스(900)는 프레임을 복수의 패치들로 세그먼트화하는 것을 포함한다. 일부 예들에서, 프레임은 정지 이미지를 포함한다. 일부 예에서, 프레임은 비디오 프레임(예를 들어, 인코딩되지 않은 비디오 프레임)을 포함한다. 일부 경우들에서, 복수의 패치들 중 각각의 패치는 균일한 사이즈를 가지며 세그먼트화된 프레임의 하나 이상의 픽셀들을 포함한다. 예를 들어, 복수의 패치들은 정지 이미지 프레임 또는 비디오 프레임을 포함하는 입력으로부터 세그먼트화될 수 있다.

[0141] 블록(904)에서, 프로세스(900)는 입력으로서 복수의 패치들을 사용하여 인코더 서브네트워크의 복수의 인코더 트랜스포머 레이어들에 의해, 인코딩된 이미지 데이터의 프레임을 생성하는 것을 포함한다. 일부 예들에서, 인코딩된 이미지 데이터의 프레임은 이미지 데이터의 잠재 표현이다. 일부 경우들에서, 잠재 표현은 인코더 서브네트워크의 복수의 인코더 트랜스포머 레이어들에 의해 생성된 계층적 특징 맵이다. 일부 예들에서, 프로세스(900)는 입력으로서 복수의 패치들을 사용하여 인코더 서브네트워크의 복수의 인코더 트랜스포머 레이어들에 의해, 세그먼트화된 프레임에 대한 계층적 특징 맵을 생성하는 것을 포함할 수 있다. 프로세스(900)는 계층적 특징 맵으로부터 인코딩된 이미지 데이터의 프레임을 생성하는 것을 포함할 수 있다.

[0142] 하나의 예시적 예에서, 인코딩된 이미지 데이터의 프레임을 생성하기 위해, 프로세서(900)는 복수의 인코더 트

랜스포머 레이어들 중 제 1 인코더 트랜스포머 레이어의 제 1 트랜스포머 블록에 의해, 하나 이상의 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것을 포함할 수 있다. 프로세스(900)는, 제 1 인코더 트랜스포머 레이어의 제 2 트랜스포머 블록에 의해, 하나 이상의 시프트된 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것을 포함할 수 있다. 일부 양태들에서, 제 1 트랜스포머 블록 및 제 2 트랜스포머 블록은 동일한 아키텍처를 갖는다. 하나 이상의 시프트된 윈도우 파티션들은 하나 이상의 윈도우 파티션들과 중첩된다. 프로세스(900)는 제 1 트랜스포머 블록 및 제 2 트랜스포머 블록 중 하나 이상에 의해, 세그먼트화된 프레임에 비선형 변환을 적용하기 위한 특징들의 하나 이상의 패치들을 결정하는 것을 더 포함할 수 있다. 프로세스(900)는 패치 병합 엔진에 의해, 제 1 인코더 트랜스포머 레이어와 제 2 인코더 트랜스포머 레이어 사이의 패치 사이즈를 증가시키는 것을 포함할 수 있다. 본 명세서에 설명된 바처럼, 패치 병합 엔진은 제 1 인코더 트랜스포머 레이어로부터의 복수의 인접한 패치들을 제 2 인코더 트랜스포머 레이어에 제공되는 병합된 패치로 결합하도록 구성된다. 일부 경우들에서, 제 1 인코더 트랜스포머 레이어의 제 2 트랜스포머 블록의 출력은 제 2 인코더 트랜스포머 레이어의 입력에 커플링된다.

[0143] 일부 경우들에서, 프로세스(900)는, 패치 병합 엔진을 사용하여, 인접한 패치들의 하나 이상의 서브세트로부터 획득된 특징들을 연쇄시킴으로써 패치 사이즈를 증가시키는 것을 포함할 수 있다. 패치 병합 엔진은 인접한 패치들의 각각의 서브세트를 병합된 패치 출력으로 병합할 수 있다.

[0144] 일부 예들에서, 프로세스(900)는 제 1 인코더 트랜스포머 레이어 이전에 인코더 서브네트워크의 선형 임베딩 레이어(예를 들어, 도 5a의 선형 임베딩 레이어(521))에 복수의 패치들을 제공하는 것을 포함할 수 있다. 선형 임베딩 레이어는 복수의 패치들에 선형 변환을 적용할 수 있다.

[0145] 일부 양태들에서, 프로세스(900)는 레이트-왜곡 손실에 적어도 부분적으로 기초한 손실 함수를 사용하여 복수의 인코더 트랜스포머 레이어들 중 하나 이상의 인코더 트랜스포머 레이어들을 훈련하는 것을 포함할 수 있다. 일부 경우들에서, 손실 함수는 레이트 왜곡을 위한 라그랑지 승수(Lagrangian multiplier)를 포함한다. 위에서 언급한 바와 같이, 레이트-왜곡 손실은 $L = D + \beta * R$ 로서 표현될 수 있다.

[0146] 일부 양태들에서, 프로세스(900)는 팩터라이즈드 프라이어를 사용하여 인코딩된 이미지 데이터를 엔트로피 코딩하는 것을 포함할 수 있다.

[0147] 일부 예들에서, 본 명세서에 설명된 프로세스들(예컨대, 프로세스(800), 프로세스(900), 및/또는 본 명세서에 설명된 임의의 다른 프로세스)은 컴퓨팅 디바이스, 장치, 또는 시스템에 의해 수행될 수도 있다. 하나의 예에서, 프로세스(800) 및/또는 프로세스(900)는 도 10의 컴퓨팅 디바이스 아키텍처(1000)를 갖는 컴퓨팅 디바이스 또는 시스템에 의해 수행될 수 있다. 컴퓨팅 디바이스, 장치 또는 시스템은 모바일 디바이스(예를 들어, 모바일 폰), 데스크탑 컴퓨팅 디바이스, 태블릿 컴퓨팅 디바이스, 웨어러블 디바이스(예를 들어, VR 헤드셋, AR 헤드셋, AR 안경, 네트워크 연결형 시계 또는 스마트워치, 또는 다른 웨어러블 디바이스), 서버 컴퓨터, 자율 주행 차량 또는 자율 주행 차량의 컴퓨팅 디바이스, 로봇 디바이스, 랩톱 컴퓨터, 스마트 텔레비전, 카메라, 및/또는 프로세스(800) 및/또는 프로세스(900) 및/또는 본 명세서에 설명된 임의의 다른 프로세스를 포함하는 본 명세서에 설명된 프로세스들을 수행하기 위한 리소스 능력들을 갖는 임의의 다른 컴퓨팅 디바이스와 같은 임의의 적합한 디바이스를 포함할 수 있다. 일부 경우에서, 컴퓨팅 디바이스 또는 장치는 하나 이상의 입력 디바이스들, 하나 이상의 출력 디바이스들, 하나 이상의 프로세서들, 하나 이상의 마이크로프로세서들, 하나 이상의 마이크로컴퓨터들, 하나 이상의 카메라들, 하나 이상의 센서들, 및/또는 본 명세서에 설명된 프로세스들의 단계들을 수행하도록 구성되는 다른 컴포넌트(들)와 같은 다양한 컴포넌트들을 포함할 수도 있다. 일부 예에서, 컴퓨팅 디바이스는 디스플레이, 데이터를 통신 및/또는 수신하도록 구성된 네트워크 인터페이스, 이들의 임의의 조합, 및/또는 다른 컴포넌트(들)를 포함할 수도 있다. 네트워크 인터페이스는 인터넷 프로토콜 (IP) 기반 데이터 또는 다른 유형의 데이터를 통신 및/또는 수신하도록 구성될 수도 있다.

[0148] 컴퓨팅 디바이스의 컴포넌트들은 회로에서 구현될 수 있다. 예를 들어, 컴포넌트들은, 하나 이상의 프로그래밍 가능 전자 회로들 (예컨대, 마이크로프로세서들, 그래픽 프로세싱 유닛들 (GPU들), 디지털 신호 프로세서들 (DSP들), 중앙 프로세싱 유닛들 (CPU들), 및/또는 다른 적합한 전자 회로들) 을 포함할 수 있는 전자 회로들 또는 다른 전자 하드웨어를 포함하거나 및/또는 이들을 사용하여 구현될 수 있거나, 및/또는 본 명세서에서 설명된 다양한 동작들을 수행하기 위해 컴퓨터 소프트웨어, 펌웨어, 또는 이들의 임의의 조합을 포함하거나 및/또는 이들을 사용하여 구현될 수 있다.

[0149] 프로세스 (800) 및 프로세스 (900) 은 논리 흐름도들로서 예시되고, 그 동작은 하드웨어, 컴퓨터 명령들, 또는

이들의 조합으로 구현될 수 있는 동작들의 시퀀스를 표현한다. 컴퓨터 명령들의 맥락에서, 그 동작들은 하나 이상의 프로세서들에 의해 실행될 때 열거된 동작들을 수행하는, 하나 이상의 컴퓨터 판독 가능 저장 매체에 저장된 컴퓨터 실행 가능 명령들을 나타낸다. 일반적으로, 컴퓨터 실행가능 명령들은, 특정 기능들을 수행하거나 또는 특정 데이터 타입들을 구현하는 루틴들, 프로그램들, 오브젝트들, 컴포넌트들, 데이터 구조들 등을 포함한다. 동작들이 설명되는 순서는 제한으로서 해석되도록 의도되지 않으며, 임의의 수의 설명된 동작들은 프로세스들을 구현하기 위해 임의의 순서로 및/또는 병렬로 결합될 수 있다.

[0150] 추가적으로, 본원에 설명된 프로세스 (800), 프로세스 (900) 및/또는 임의의 다른 프로세스는 실행가능 명령들로 구성된 하나 이상의 컴퓨터 시스템들의 제어 하에서 수행될 수도 있고, 집합적으로 하나 이상의 프로세서 상에서 실행하는 코드 (예를 들어, 실행가능 명령들, 하나 이상의 컴퓨터 프로그램들, 또는 하나 이상의 애플리케이션들)로서, 하드웨어에 의해, 또는 이들의 조합으로 구현될 수도 있다. 전술한 바와 같이, 코드는 예를 들어, 하나 이상의 프로세서에 의해 실행 가능한 복수의 명령을 포함하는 컴퓨터 프로그램의 형태로 컴퓨터 판독 가능 또는 머신 판독 가능 저장 매체에 저장될 수도 있다. 컴퓨터 판독 가능 또는 머신 판독 가능 저장 매체는 비일시적일 수도 있다.

[0151] 도 10 는 본 명세서에 설명된 다양한 기법을 구현할 수 있는 예시적인 컴퓨팅 디바이스의 예시적인 컴퓨팅 디바이스 아키텍처(1000)를 예시한다. 일부 예들에서, 컴퓨팅 디바이스는 모바일 디바이스, 웨어러블 디바이스, 확장 현실 디바이스 (예를 들어, 가상 현실 (VR) 디바이스, 증강 현실 (AR) 디바이스 또는 혼합 현실 (MR) 디바이스), 퍼스널 컴퓨터, 랩톱 컴퓨터, 비디오 서버, 차량 (또는 차량의 컴퓨팅 디바이스) 또는 다른 디바이스를 포함할 수 있다. 예를 들어, 컴퓨팅 디바이스 아키텍처(1000)는 도 6a 의 시스템을 구현할 수 있다. 컴퓨팅 디바이스 아키텍처(1000)의 컴포넌트들은 버스와 같은 연결(1005)을 사용하여 서로 전기적으로 통신하는 것으로 도시된다. 예시적인 컴퓨팅 디바이스 아키텍처(1000)는 프로세싱 유닛(CPU 또는 프로세서)(1010), 및 판독 전용 메모리(ROM)(1020) 및 랜덤 액세스 메모리(RAM)(1025)와 같은 컴퓨팅 디바이스 메모리(1015)를 포함하는 다양한 컴퓨팅 디바이스 컴포넌트들을 프로세서(1010)에 커플링하는 컴퓨팅 디바이스 연결부(1005)를 포함한다.

[0152] 컴퓨팅 디바이스 아키텍처(1000)는 프로세서(1010)와 직접 연결되거나, 그에 근접하거나, 또는 그의 일부로서 통합된 고속 메모리의 캐시를 포함할 수 있다. 컴퓨팅 디바이스 아키텍처 (1000) 는 프로세서 (1010) 에 의한 빠른 액세스를 위해 메모리 (1015) 및/또는 저장 디바이스 (1030)로부터 캐시 (1012)로 데이터를 복사할 수 있다. 이러한 방식으로, 캐시는 데이터를 기다리는 동안 프로세서 (1010) 지연을 방지하는 성능 부스트를 제공할 수 있다. 이들 및 다른 모듈들은 다양한 액션들을 수행하도록 프로세서 (1010) 를 제어하거나 제어하도록 구성될 수 있다. 다른 컴퓨팅 디바이스 메모리 (1015) 가 물론 사용을 위해 이용가능할 수도 있다. 메모리 (1015) 는 상이한 성능 특성들을 갖는 다수의 상이한 타입들의 메모리를 포함할 수 있다. 프로세서 (1010) 는, 임의의 범용 프로세서 및 프로세서 (1010) 를 제어하도록 구성된, 저장 디바이스(1030)에 저장된, 서비스 1 (1032), 서비스 2 (1034), 및 서비스 3 (1036)과 같은, 하드웨어 또는 소프트웨어 서비스뿐만 아니라, 소프트웨어 명령들이 프로세서 설계에 통합되는 특수 목적 프로세서를 포함할 수 있다. 프로세서 (1010) 는 다수의 코어들 또는 프로세서들, 버스, 메모리 제어기, 캐시 등을 포함하는 독립형 시스템일 수도 있다. 다중 코어 프로세서는 대칭형 또는 비대칭형일 수도 있다.

[0153] 컴퓨팅 디바이스 아키텍처 (1000) 와의 사용자 상호작용을 가능하게 하기 위해, 입력 디바이스 (1045) 는 스피치를 위한 마이크, 제스처 또는 그래픽 입력을 위한 터치 감지 스크린, 키보드, 마우스, 모션 입력, 스피치 등과 같은 임의의 수의 입력 메커니즘을 나타낼 수 있다. 출력 디바이스 (1035) 는 또한 디스플레이, 프로젝터, 텔레비전, 스피커 디바이스 등과 같이 당업자에게 알려진 다수의 출력 메커니즘 중 하나 이상일 수 있다. 일부 경우에, 다중모드 컴퓨팅 디바이스들은 사용자가 컴퓨팅 디바이스 아키텍처 (1000) 와 통신하기 위해 여러 유형의 입력을 제공하는 것을 가능하게 할 수 있다. 통신 인터페이스 (1040) 는 일반적으로 사용자 입력 및 컴퓨팅 디바이스 출력을 지배하고 관리할 수 있다. 임의의 특정 하드웨어 배열에 대해 동작하는 것에 대한 제한은 없으며, 따라서 여기서 기본 특징들은 이들이 개발됨에 따라 개선된 하드웨어 또는 펌웨어 배열들을 쉽게 대체할 수도 있다.

[0154] 저장 디바이스 (1030) 는 비휘발성 메모리이고, 하드 디스크 또는 컴퓨터에 의해 액세스가능한 데이터를 저장할 수 있는 다른 유형의 컴퓨터 판독가능 매체, 이를테면 자기 카세트, 플래시 메모리 카드, 솔리드 스테이트 메모리 디바이스, 디지털 다기능 디스크, 카트리지, 랜덤 액세스 메모리들 (RAM들) (1025), 판독 전용 메모리 (ROM) (1020) 및 이들의 하이브리드일 수 있다. 저장 디바이스 (1030) 는 프로세서 (1010) 를 제어하기 위한 서비스들 (1032, 1034, 1036) 을 포함할 수 있다. 다른 하드웨어 또는 소프트웨어 모듈들 또는 엔진들이 고려된다. 저장 디바이스(1030)는 컴퓨팅 디바이스 연결부(1005)에 연결될 수 있다. 일 양태에서, 특정 기능을 수행하는 하드웨

어 모듈은, 그 기능을 수행하기 위해, 프로세서 (1010), 연결부 (1005), 출력 디바이스 (1035) 등과 같은 필요한 하드웨어 컴포넌트들과 관련하여 컴퓨터 판독가능 매체에 저장된 소프트웨어 컴포넌트를 포함할 수 있다.

[0155] 본 개시의 양태들은 하나 이상의 능동 심도 감지 시스템들을 포함하거나 그들에 커플링된 (보안 시스템들, 스마트폰들, 태블릿들, 랩탑 컴퓨터들, 차량들, 드론들, 또는 다른 디바이스들과 같은) 임의의 적합한 전자 디바이스에 적용가능하다. 하나의 광 프로젝터를 갖거나 그에 커플링된 디바이스에 관하여 하기에서 설명되지만, 본 개시의 양태들은 임의의 수의 광 프로젝터들을 갖는 디바이스들에 적용가능하고, 따라서, 특정 디바이스들로 제한되지 않는다.

[0156] 용어 "디바이스" 는 (하나의 스마트폰, 하나의 제어기, 하나의 프로세싱 시스템 등과 같은) 하나 또는 특정 수의 물리적 오브젝트들로 제한되지 않는다. 본 명세서에서 사용되는 바와 같이, 디바이스는 본 개시의 적어도 일부 부분들을 구현할 수도 있는 하나 이상의 부분들을 갖는 임의의 전자 디바이스일 수도 있다. 하기의 설명 및 예들이 본 개시의 다양한 양태들을 설명하기 위해 용어 "디바이스" 를 사용하지만, 용어 "디바이스" 는 오브젝트들의 특정 구성, 타입, 또는 개수로 제한되지 않는다. 부가적으로, 용어 "시스템" 은 다수의 컴포넌트들 또는 특정 실시형태들로 제한되지 않는다. 예를 들어, 시스템은 하나 이상의 인쇄 회로 보드들 또는 다른 기판들 상에서 구현될 수도 있고, 이동가능 또는 정적 컴포넌트들을 가질 수도 있다. 하기의 설명 및 예들이 본 개시의 다양한 양태들을 설명하기 위해 용어 "시스템" 을 사용하지만, 용어 "시스템" 은 오브젝트들의 특정 구성, 타입, 또는 개수로 제한되지 않는다.

[0157] 구체적 상세들은 본원에 제공된 실시형태들 및 예들의 철저한 이해를 제공하기 위하여 상기 설명에서 제공되었다. 하지만, 실시형태들은 이들 특정 상세들 없이 실시될 수도 있음이 당업자에 의해 이해될 것이다. 설명의 명료성을 위해, 일부 사례들에서, 본 기술은 디바이스들, 디바이스 컴포넌트들, 소프트웨어로 구체화된 방법에서의 단계들 또는 루틴들, 또는 하드웨어와 소프트웨어의 조합들을 포함하는 기능적 블록들을 포함하는 개개의 기능적 블록들을 포함하는 것으로서 제시될 수도 있다. 도면들에서 도시된 그리고/또는 본 명세서에서 설명된 것들 이외의 추가적인 컴포넌트들이 사용될 수도 있다. 예를 들어, 회로들, 시스템들, 네트워크들, 프로세스들 및 다른 컴포넌트들은, 불필요한 상세로 실시형태들을 모호하게 하지 않기 위해 블록도 형태에서 컴포넌트들로서 도시될 수도 있다. 다른 예들에서, 잘 알려진 회로들, 프로세스들, 알고리즘들, 구조들, 및 기법들은 실시형태들을 모호하게 하는 것을 회피하기 위해 불필요한 상세 없이 도시될 수도 있다.

[0158] 개별 실시형태들은, 플로우차트, 흐름도, 데이터 흐름도, 구조도, 또는 블록도로서 도시되는 프로세스 또는 방법으로서 위에서 설명될 수도 있다. 비록 플로우차트가 동작들을 순차적인 프로세스로서 설명할 수도 있지만, 동작들 중 다수는 병렬로 또는 동시에 수행될 수 있다. 또한, 동작들의 순서는 재배열될 수도 있다. 프로세스는 그의 동작들이 완료되면 종결되지만, 도면에 포함되지 않은 추가의 단계들을 가질 수 있다. 프로세스는 방법 (method), 함수 (function), 프로시저 (procedure), 서브루틴 (subroutine), 서브프로그램 (subprogram) 등에 대응할 수도 있다. 프로세스가 함수에 대응할 때, 그의 종결은 호출 함수 (calling function) 또는 메인 함수 (main function) 에 대한 함수의 리턴에 대응할 수 있다.

[0159] 상술한 예들에 따른 프로세스들 및 방법들은 컴퓨터 판독가능 매체들로부터 저장되거나 그렇지 않으면 이용가능한 컴퓨터 실행가능 명령들을 사용하여 구현될 수 있다. 그러한 명령들은 예를 들어, 범용 컴퓨터, 특수 목적 컴퓨터, 또는 프로세싱 디바이스가 특정 기능 또는 기능들의 그룹을 수행하게 하거나 그렇지 않으면 수행하도록 구성하는 명령들 및 데이터를 포함할 수 있다. 사용된 컴퓨터 리소스들의 부분들은 네트워크를 통해 액세스가능할 수 있다. 컴퓨터 실행 가능 명령들은 예를 들어 이진, 중간 형식 명령 이라면 어셈블리 언어, 펌웨어, 소스 코드 등일 수도 있다.

[0160] 용어 "컴퓨터 판독가능 매체" 는, 휴대 또는 비휴대 저장 디바이스, 광학 저장 디바이스, 및 명령(들) 및/또는 데이터를 저장, 포함 또는 나눌 수 있는 다양한 다른 매체를 포함하지만, 이에 한정되지는 않는다. 컴퓨터 판독가능 매체는 데이터가 저장될 수 있고 반송파 및/또는 무선 또는 유선 접속을 통해 전파되는 일시적 전자 신호를 포함하지 않는 비일시적 매체를 포함할 수도 있다. 비일시적 매체의 예는 다른 것들 중에서도, 자기 디스크 또는 테이프, 플래시 메모리와 같은 광학 저장 매체, 메모리 또는 메모리 디바이스, 자기 또는 광 디스크, 플래시 메모리, 비휘발성 메모리가 제공된 USB 디바이스, 네트워킹된 저장 디바이스, CD(Compact Disk) 또는 DVD(Digital Versatile Disk), 이들의 적절한 조합 등을 포함할 수도 있지만 이에 제한되지 않는다. 컴퓨터 판독가능 매체는 프로시저, 함수, 서브프로그램, 프로그램, 루틴, 서브루틴, 모듈, 엔진, 소프트웨어 패키지, 클래스, 또는 명령들, 데이터 구조들, 또는 프로그램 스테이트먼트들의 임의의 조합을 나타낼 수도 있는 코드 및/또는 머신 실행가능 명령들을 저장했을 수도 있다. 코드 세그먼트는, 정보, 데이터, 인수들 (arguments), 파라

미터들, 또는 메모리 콘텐츠들을 전달 및/또는 수신함으로써 다른 코드 세그먼트 또는 하드웨어 회로에 커플링될 수도 있다. 정보, 인수들, 파라미터들, 데이터 등은 메모리 공유, 메시지 전달, 토큰 전달, 네트워크 송신 등을 포함한 임의의 적합한 수단을 통해 전달, 포워딩, 또는 송신될 수도 있다.

[0161] 일부 실시형태들에서, 컴퓨터 판독가능 저장 디바이스들, 매체들, 및 메모리들은 비트 스트림 등을 포함하는 무선 신호 또는 케이블을 포함할 수 있다. 하지만, 언급될 때, 비일시적 컴퓨터 판독가능 저장 매체들은 에너지, 캐리어 신호들, 전자기 파들, 및 신호들 그 자체와 같은 매체들을 명시적으로 배제한다.

[0162] 이들 개시에 따른 프로세스들 및 방법들을 구현하는 디바이스들은 하드웨어, 소프트웨어, 펌웨어, 미들웨어, 마이크로코드, 하드웨어 기술 언어, 또는 이들의 임의의 조합을 포함할 수 있고, 다양한 폼 팩터들 중 임의의 것을 취할 수 있다. 소프트웨어, 펌웨어, 미들웨어, 또는 마이크로코드로 구현될 경우, 필요한 작업들을 수행하기 위한 프로그램 코드 또는 코드 세그먼트들 (예를 들어, 컴퓨터 프로그램 제품)은 컴퓨터 판독가능 또는 머신 판독가능 매체에 저장될 수도 있다. 프로세서(들)는 필요한 작업들을 수행할 수도 있다. 폼 팩터들의 통상적인 예들은 랩톱, 스마트 폰, 모바일 폰, 태블릿 디바이스 또는 다른 소형 폼 팩터 개인용 컴퓨터, PDA(personal digital assistant), 랙마운트 디바이스, 독립적인 디바이스 등을 포함한다. 본 명세서에 설명된 기능성은 또한 주변기기 또는 애드-인 카드 (add-in card) 에서 구체화될 수 있다. 이러한 기능성은 또한, 추가적인 예로서, 단일 디바이스에서 실행되는 상이한 칩들 또는 상이한 프로세스들 사이에서 회로 보드 상에서 구현될 수 있다.

[0163] 명령들, 이러한 명령들을 운반하기 위한 매체들, 그것들을 실행하기 위한 컴퓨팅 리소스들, 및 이러한 컴퓨팅 리소스들을 지원하기 위한 다른 구조들은 본 개시에서 설명된 기능들을 제공하기 위한 예시적인 수단들이다.

[0164] 전술한 설명에 있어서, 본 출원의 양태들은 그의 특정 실시형태들을 참조하여 설명되지만, 당업자는 본 출원이 그에 한정되지 않음을 인식할 것이다. 따라서, 본원의 예시적 실시 형태들이 본원에서 상세하게 설명되었지만, 본 발명의 개념은 이와 달리 다양하게 구체화되고 채용될 수도 있으며, 첨부된 청구 범위는 선행 기술에 의해 제한되는 것을 제외하고는 그러한 변형을 포함하는 것으로 해석되도록 의도된다. 전술한 애플리케이션의 다양한 특징들 및 양태들은 개별적으로 또는 공동으로 사용될 수도 있다. 추가로, 실시형태들은 본 명세서의 더 넓은 사상 및 범위로부터 일탈함없이 본 명세서에서 설명된 것들을 넘어서는 임의의 수의 환경들 및 애플리케이션들에서 활용될 수 있다. 본 명세서 및 도면들은, 이에 따라, 제한적이기 보다는 예시적인 것으로서 간주되어야 한다. 예시의 목적들을 위해, 방법들은 특정 순서로 설명되었다. 대안적인 실시형태들에서, 방법들은 설명된 것과 다른 상이한 순서로 수행될 수도 있음이 인식되어야 한다.

[0165] 당업자는 본 명세서에서 사용된 미만 (" $<$ ") 및 초과 (" $>$ ") 기호들 또는 용어가 본 설명의 범위로부터 일탈함 없이, 각각 이하 (" $\leq$ ") 및 이상 (" $\geq$ ") 기호들로 대체될 수 있음을 알 것이다.

[0166] 컴포넌트들이 특정 동작을 수행하도록 "구성된" 것으로 기술되는 경우, 그러한 구성은 예를 들어, 전자 회로 또는 다른 하드웨어를 설계하여 그 동작을 수행하는 것에 의해, 프로그래머블 전자 회로 (예를 들어, 마이크로프로세서 또는 다른 적절한 전자 회로)를 프로그래밍하여 그 동작을 수행하는 것에 의해 또는 이들의 임의의 조합에 의해, 달성될 수 있다.

[0167] 어구 "에 커플링된 (coupled to)"은 직접 또는 간접적으로 중 어느 하나로 다른 컴포넌트에 물리적으로 연결된 임의의 컴포넌트, 및/또는 직접 또는 간접적으로 중 어느 하나로 다른 컴포넌트와 통신하는 (예컨대, 유선 또는 무선 연결, 및/또는 다른 적합한 통신 인터페이스를 통해 다른 컴포넌트에 연결된) 임의의 컴포넌트를 지칭한다.

[0168] 세트 "중 적어도 하나" 또는 세트 "중 하나 이상"을 인용하는 청구항 문언 또는 다른 문언은 그 세트의 하나의 멤버 또는 그 세트의 다수의 멤버들 (임의의 조합)이 청구항을 충족하는 것을 나타낸다. 예를 들어, "A 및 B 중 적어도 하나" 또는 "A 또는 B 중 적어도 하나"를 인용하는 청구항 문언은 A, B, 또는 A 및 B를 의미한다. 다른 예에서, "A, B, 및 C 중 적어도 하나" 또는 "A, B, 또는 C 중 적어도 하나"를 인용하는 청구항 문언은 A, B, C, 또는 A 및 B, 또는 A 및 C, 또는 B 및 C, 또는 A 및 B 및 C를 의미한다. 문언 세트 "중 적어도 하나" 및/또는 세트 중 "하나 이상"은 세트를 그 세트에 열거된 항목들로 제한하지 않는다. 예를 들어, "A 및 B 중 적어도 하나" 또는 "A 또는 B 중 적어도 하나"를 인용하는 청구항 문언은 A, B, 또는 A 및 B를 의미할 수 있으며, A 및 B의 세트에 열거되지 않은 항목들을 추가적으로 포함할 수 있다.

[0169] 본 명세서에 개시된 실시형태들과 관련하여 설명된 다양한 예시적인 논리 블록들, 모듈들, 엔진들, 회로들, 및 알고리즘 단계들은 전자 하드웨어, 컴퓨터 소프트웨어, 펌웨어, 또는 이들의 조합들로서 구현될 수도 있다. 하드웨어와 소프트웨어의 이러한 상호교환가능성을 명확히 예시하기 위해, 다양한 예시적인 컴포넌트, 블록,

모듈, 엔진, 회로, 및 단계가 일반적으로 그들의 기능성의 관점에서 위에서 설명되었다. 그러한 기능성이 하드웨어 또는 소프트웨어로 구현될지 여부는, 전체 시스템에 부과된 설계 제약 및 특정 애플리케이션에 의존한다. 당업자는 설명된 기능성을 각각의 특정 애플리케이션에 대하여 다양한 방식으로 구현할 수도 있지만, 그러한 구현 결정들이 본 출원의 범위로부터의 이탈을 야기하는 것으로서 해석되지는 않아야 한다.

[0170] 본원에 기술된 기법들은 또한 전자 하드웨어, 컴퓨터 소프트웨어, 펌웨어 또는 이들의 임의의 조합으로 구현될 수도 있다. 그러한 기술들은 범용 컴퓨터들, 무선 통신 디바이스 핸드셋들, 또는 무선 통신 디바이스 핸드셋들 및 다른 디바이스들에서의 애플리케이션을 포함하여 다수의 이용들을 갖는 집적 회로 디바이스들과 같은 임의의 다양한 디바이스들에서 구현될 수도 있다. 모듈들, 또는 컴포넌트들로서 설명된 임의의 피쳐들은 집적 로직 디바이스 (integrated logic device) 에서 함께 구현되거나 또는 이산이지만 연동적인 (interoperable) 로직 디바이스들로서 따로 구현될 수도 있다. 소프트웨어로 구현되면, 기법들은, 실행될 때, 위에서 설명된 방법들 중 하나 이상을 수행하는 명령들을 포함하는 프로그램 코드를 포함하는 컴퓨터 판독가능 데이터 저장 매체에 의해 적어도 부분적으로 실현될 수도 있다. 컴퓨터 판독가능 데이터 저장 매체는 컴퓨터 프로그램 제품의 일부를 형성할 수도 있으며, 이는 패키징 재료들을 포함할 수도 있다. 컴퓨터 판독가능 매체는 동적 랜덤 액세스 메모리 (SDRAM) 와 같은 랜덤 액세스 메모리 (RAM), 판독 전용 메모리 (ROM), 비휘발성 랜덤 액세스 메모리 (NVRAM), 전기적으로 소거가능한 프로그래머블 판독 전용 메모리 (EEPROM), 플래시 메모리, 자기 또는 광학 데이터 저장 매체들 등과 같은 메모리 또는 데이터 저장 매체들을 포함할 수도 있다. 추가로 또는 대안적으로 기법들은, 전파된 신호들 또는 파들과 같은, 명령들 또는 데이터 구조들의 형태의 프로그램 코드를 나르거나 통신하고, 컴퓨터에 의해 액세스, 판독 및/또는 실행될 수 있는 컴퓨터 판독가능 통신 매체에 의해 적어도 부분적으로 실현될 수도 있다.

[0171] 프로그램 코드는 프로세서에 의해 실행될 수도 있고, 프로세서는 하나 이상의 프로세서들, 이를테면 하나 이상의 DSP (digital signal processor), 범용 마이크로프로세서, ASIC (application specific integrated circuit), FPGA (field programmable logic array), 또는 다른 등가 집적 또는 이산 로직 회로를 포함할 수도 있다. 그러한 프로세서는 본 개시에서 설명된 기법들 중 임의의 기법을 수행하도록 구성될 수도 있다. 범용 프로세서는 마이크로프로세서일 수도 있지만, 대안으로, 프로세서는 임의의 종래의 프로세서, 제어기, 마이크로제어기, 또는 상태 머신일 수도 있다. 프로세서는 또한, 컴퓨팅 디바이스들의 조합, 예를 들어, DSP 와 마이크로프로세서의 조합, 복수의 마이크로프로세서들, DSP 코어와 결합된 하나 이상의 마이크로프로세서들, 또는 임의의 다른 이러한 구성으로서 구현될 수도 있다. 따라서, 본 명세서에서 사용된 바와 같은 용어 "프로세서" 는 전술한 구조, 전술한 구조의 임의의 조합, 또는 본 명세서에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 또는 장치 중 임의의 것을 지칭할 수도 있다.

[0172] 본 개시의 예시적 양태들은 다음을 포함한다:

[0173] 양태 1: 미디어 데이터를 프로세싱하기 위한 장치로서, 적어도 하나의 메모리; 및 상기 적어도 하나의 메모리에 커플링된 적어도 하나의 프로세서를 포함하고, 상기 적어도 하나의 프로세서는: 인코딩된 이미지 데이터의 프레임 임의 잠재 표현(latent representation)을 획득하고; 입력으로서 상기 인코딩된 이미지 데이터의 프레임의 잠재 표현을 사용하여 디코더 서브네트워크의 복수의 디코더 트랜스포머 레이어들에 기초하여, 디코딩된 이미지 데이터의 프레임을 생성하도록 구성되고, 상기 복수의 디코더 트랜스포머 레이어들 중 적어도 하나의 디코더 트랜스포머 레이어는 특징들의 하나 이상의 패치들을 생성하고 상기 하나 이상의 패치들 상에 적용된 하나 이상의 윈도우 파티션들 및 시프트된 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하도록 구성된 하나 이상의 트랜스포머 블록들; 및 상기 하나 이상의 패치들 중 각각의 패치의 개개의 사이즈를 줄이도록 구성되는 패치 병합 해제 엔진을 포함하는, 미디어 데이터를 프로세싱하기 위한 장치.

[0174] 양태 2: 양태 1에 있어서, 상기 디코딩된 이미지 데이터의 프레임을 생성하기 위해, 상기 적어도 하나의 프로세서는 상기 복수의 디코더 트랜스포머 레이어들 중 제 1 디코더 트랜스포머 레이어의 제 1 트랜스포머 블록을 사용하여, 상기 하나 이상의 패치들 상에 적용된 하나 이상의 제 1 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하고; 상기 제 1 디코더 트랜스포머 레이어의 제 2 트랜스포머 블록을 사용하여, 상기 하나 이상의 패치들 상에 적용된 하나 이상의 제 2 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것으로서, 상기 하나 이상의 제 2 윈도우 파티션들은 상기 하나 이상의 제 1 윈도우 파티션들 중 인접한 것들 사이에 하나 이상의 경계들과 중첩되도록 시프트되는, 상기 셀프 어텐션을 결정하고; 상기 패치 병합 해제 엔진을 사용하여, 복수의 병합 해제된 패치들로 상기 하나 이상의 패치들의 각각의 패치를 세그먼트화하는 것으로서, 상기 복수의 병합 해제된 패치들은 중첩되지 않는, 상기 하나 이상의 패치들의 각각의 패치를 세그먼트화하도록 구성되는,

미디어 데이터를 프로세싱하기 위한 장치.

- [0175] 양태 3: 양태 2에 있어서, 상기 적어도 하나의 프로세서는 상기 복수의 디코더 트랜스포머 레이어들 중 제 2 디코더 트랜스포머 레이어의 제 1 트랜스포머 블록에 상기 복수의 병합 해제된 패치들을 제공하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0176] 양태 4: 양태 3에 있어서, 상기 적어도 하나의 프로세서는 다음과 같이 구성된다: 상기 제 2 디코더 트랜스포머 레이어의 패치 병합 해제 엔진을 사용하여, 상기 복수의 병합 해제된 패치들을 세그먼트화하고; 상기 패치 병합 해제 엔진의 출력을 상기 복수의 디코더 트랜스포머 레이어들 중 제 3 디코더 트랜스포머 레이어에 제공하는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0177] 양태 5: 양태 2 내지 4 중 어느 것에 있어서, 상기 복수의 병합 해제된 패치들 중 각각의 병합 해제된 패치는 균일한 패치 사이즈를 갖고 상기 패치 병합 해제 엔진은 2의 패치 사이즈 감소 팩터를 적용하는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0178] 양태 6: 양태 2 내지 5 중 어느 것에 있어서, 복수의 병합 해제된 패치들로 상기 하나 이상의 패치들 중 각각의 패치를 세그먼트화하기 위해, 상기 적어도 하나의 프로세서는 상기 복수의 병합 해제된 패치들의 특징 차원(feature dimension)을 감소시키도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0179] 양태 7: 양태 1 내지 6 중 어느 것에 있어서, 상기 복수의 디코더 트랜스포머 레이어들은 상기 인코딩된 이미지 데이터의 프레임의 잠재 표현을 입력으로 수신하고 비선형 변환을 적용하여 디코딩된 이미지 데이터의 프레임을 생성하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0180] 양태 8: 양태 7에 있어서, 상기 비선형 변환은 합성 변환이고, 상기 디코딩된 이미지 데이터의 프레임은 상기 인코딩된 이미지 데이터의 프레임과 연관된 입력 이미지의 재구성인, 미디어 데이터를 프로세싱하기 위한 장치.
- [0181] 양태 9: 양태 1 내지 8 중 어느 것에 있어서, 상기 적어도 하나의 프로세서는 레이트-왜곡(rate-distortion)에 적어도 부분적으로 기초한 손실 함수를 사용하여 상기 복수의 디코더 트랜스포머 레이어들 중 하나 이상의 디코더 트랜스포머 레이어들을 훈련하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0182] 양태 10: 양태 9에 있어서, 상기 손실 함수는 레이트 왜곡을 위한 라그랑지 승수(Lagrangian multiplier)를 포함하는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0183] 양태 11: 양태 1 내지 10 중 어느 것에 있어서, 상기 적어도 하나의 디코더 트랜스포머 레이어에 포함된 상기 하나 이상의 트랜스포머 블록들 중 적어도 일부는 동일한 아키텍처를 갖는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0184] 양태 12: 양태 1 내지 11 중 어느 것에 있어서, 상기 적어도 하나의 디코더 트랜스포머 레이어에 포함된 상기 하나 이상의 트랜스포머 블록들의 각각은 동일한 아키텍처를 갖는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0185] 양태 13: 양태 1 내지 12 중 어느 것에 있어서, 상기 인코딩된 이미지 데이터의 프레임은 인코딩된 정지 이미지를 포함하는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0186] 양태 14: 양태 1 내지 13 중 어느 것에 있어서, 상기 인코딩된 이미지 데이터의 프레임은 인코딩된 비디오 프레임을 포함하는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0187] 양태 15: 양태 1 내지 14 중 어느 것에 있어서, 상기 적어도 하나의 프로세서는 적어도 제 1 훈련 데이터 세트 및 제 2 훈련 데이터 세트로 상기 복수의 디코더 트랜스포머 레이어들을 훈련하도록 구성되고, 상기 제 2 훈련 데이터 세트의 데이터는 상기 제 1 훈련 데이터 세트의 데이터와 비교하여 역 시간 순서(reversed temporal order)를 갖는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0188] 양태 16: 양태 1 내지 15 중 어느 것에 있어서, 상기 복수의 디코더 트랜스포머 레이어들은 일련의 연속적인 디코더 트랜스포머 레이어들을 포함하는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0189] 양태 17: 미디어 데이터를 프로세싱하기 위한 장치로서, 적어도 하나의 메모리; 및 상기 적어도 하나의 메모리에 커플링된 적어도 하나의 프로세서를 포함하고, 상기 적어도 하나의 프로세서는: 프레임을 복수의 패치들로 세그먼트화하고; 입력으로서 상기 복수의 패치들을 사용하여 인코더 서브네트워크의 복수의 인코더 트랜스포머 레이어들에 기초하여, 인코딩된 이미지 데이터의 프레임을 생성하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.

- [0190] 양태 18: 양태 17에 있어서, 상기 인코딩된 이미지 데이터의 프레임을 생성하기 위해, 상기 적어도 하나의 프로세서는 상기 복수의 인코더 트랜스포머 레이어들 중 제 1 인코더 트랜스포머 레이어의 제 1 트랜스포머 블록을 사용하여, 하나 이상의 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하고; 상기 제 1 인코더 트랜스포머 레이어의 제 2 트랜스포머 블록을 사용하여, 하나 이상의 시프트된 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것으로서, 상기 하나 이상의 시프트된 윈도우 파티션들은 상기 하나 이상의 윈도우 파티션들과 중첩하는, 상기 셀프 어텐션을 결정하고; 상기 제 1 트랜스포머 블록 및 상기 제 2 트랜스포머 블록 중 하나 이상을 사용하여, 상기 세그먼트화된 프레임에 비선형 변환을 적용하기 위한 특징들의 하나 이상의 패치들을 결정하고; 패치 병합 엔진을 사용하여, 상기 제 1 인코더 트랜스포머 레이어와 제 2 인코더 트랜스포머 레이어 사이의 패치 사이즈를 증가시키도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0191] 양태 19: 양태 18에 있어서, 상기 패치 병합 엔진은 상기 제 1 인코더 트랜스포머 레이어로부터의 복수의 인접한 패치들을 상기 제 2 인코더 트랜스포머 레이어에 제공되는 병합된 패치로 결합하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0192] 양태 20: 양태 18 또는 양태 19 중 어느 것에 있어서, 상기 제 1 인코더 트랜스포머 레이어의 상기 제 2 트랜스포머 블록의 출력은 상기 제 2 인코더 트랜스포머 레이어의 입력에 커플링되는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0193] 양태 21: 양태 18 내지 20 중 어느 것에 있어서, 상기 적어도 하나의 프로세서는 입력으로서 상기 복수의 패치들을 사용하여 상기 인코더 서브네트워크의 상기 복수의 인코더 트랜스포머 레이어들을 사용하여, 상기 세그먼트화된 프레임에 대한 계층적 특징 맵을 생성하고; 상기 계층적 특징 맵으로부터 상기 인코딩된 이미지 데이터의 프레임을 생성하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0194] 양태 22: 양태 18 내지 21 중 어느 것에 있어서, 상기 복수의 패치들 중 각각의 패치는 균일한 사이즈를 가지며 상기 세그먼트화된 프레임의 하나 이상의 픽셀들을 포함하는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0195] 양태 23: 양태 18 내지 22 중 어느 것에 있어서, 상기 패치 병합 엔진은, 인접한 패치들의 하나 이상의 서브세트들에서 얻은 특징들을 연쇄(concatenating)시켜 상기 패치 사이즈를 늘리도록 구성되고, 상기 인접한 패치들의 각각의 서브세트는 상기 패치 병합 엔진에 의해 병합된 패치 출력으로 병합되는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0196] 양태 24: 양태 18 내지 23 중 어느 것에 있어서, 상기 제 1 트랜스포머 블록 및 상기 제 2 트랜스포머 블록은 동일한 아키텍처를 갖는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0197] 양태 25: 양태 18 내지 24 중 어느 것에 있어서, 상기 적어도 하나의 프로세서는 상기 제 1 인코더 트랜스포머 레이어 이전에 상기 인코더 서브네트워크의 선형 임베딩 레이어에 상기 복수의 패치들을 제공하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0198] 양태 26: 양태 17 내지 25 중 어느 것에 있어서, 상기 인코딩된 이미지 데이터의 프레임은 이미지 데이터의 잠재 표현인, 미디어 데이터를 프로세싱하기 위한 장치.
- [0199] 양태 27: 양태 26에 있어서, 상기 잠재 표현은 상기 인코더 서브네트워크의 상기 복수의 인코더 트랜스포머 레이어들에 의해 생성된 계층적 특징 맵인, 미디어 데이터를 프로세싱하기 위한 장치.
- [0200] 양태 28: 양태 17 내지 27 중 어느 것에 있어서, 상기 적어도 하나의 프로세서는 레이트-왜곡 손실에 기초한 손실 함수를 사용하여 상기 복수의 인코더 트랜스포머 레이어들 중 하나 이상의 인코더 트랜스포머 레이어들을 훈련하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0201] 양태 29: 양태 28에 있어서, 상기 손실 함수는 레이트 왜곡을 위한 라그랑지 승수(Lagrangian multiplier)를 포함하는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0202] 양태 30: 양태 17 내지 29 중 어느 것에 있어서, 상기 복수의 패치들은 정지 이미지 프레임 또는 비디오 프레임을 포함하는 입력으로부터 세그먼트화되는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0203] 양태 31: 양태 17 내지 30 중 어느 것에 있어서, 상기 적어도 하나의 프로세서는 팩터라이즈드 프라이어를 이용하여 상기 인코딩된 이미지 데이터를 엔트로피 코딩하도록 구성되는, 미디어 데이터를 프로세싱하기 위한 장치.
- [0204] 양태 32: 미디어 데이터를 프로세싱하는 방법으로서, 인코딩된 이미지 데이터의 프레임의 잠재 표현(latent representation)을 획득하는 단계; 및 입력으로서 상기 인코딩된 이미지 데이터의 프레임의 잠재 표현을 사용하

여 디코더 서브네트워크의 복수의 디코더 트랜스포머 레이어들에 의해, 디코딩된 이미지 데이터의 프레임을 생성하는 단계를 포함하고, 상기 복수의 디코더 트랜스포머 레이어들 중 적어도 하나의 디코더 트랜스포머 레이어는 특징들의 하나 이상의 패치들을 생성하고 상기 하나 이상의 패치들 상에 적용된 하나 이상의 윈도우 파티션들 및 시프트된 윈도우 파티션들 내에서 국부적으로 셀프 어텐션(self-attention)을 결정하기 위한 하나 이상의 트랜스포머 블록들; 및 하나 이상의 패치들 중 각각의 패치의 개개의 사이즈를 줄이기 위한 패치 병합 해제 엔진(patch un-merging engine)을 포함하는, 미디어 데이터를 프로세싱하는 방법.

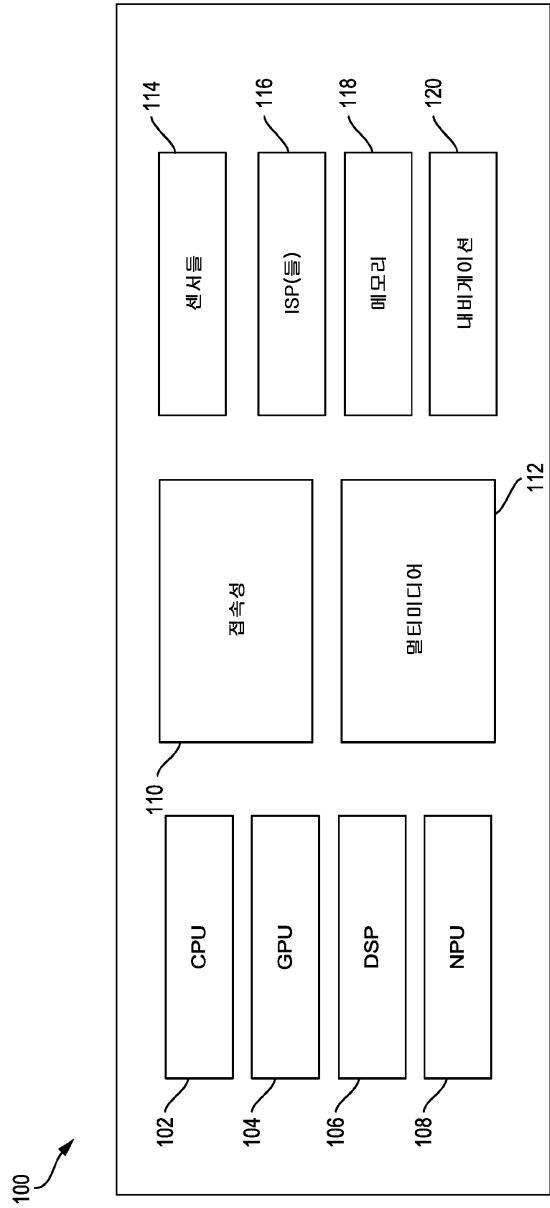
- [0205] 양태 33: 양태 32에 있어서, 상기 복수의 디코더 트랜스포머 레이어들에 의해, 상기 디코딩된 이미지 데이터의 프레임을 생성하는 단계는 상기 복수의 디코더 트랜스포머 레이어들 중 제 1 디코더 트랜스포머 레이어의 제 1 트랜스포머 블록에 의해, 상기 하나 이상의 패치들 상에 적용된 하나 이상의 제 1 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것; 상기 제 1 디코더 트랜스포머 레이어의 제 2 트랜스포머 블록에 의해, 상기 하나 이상의 패치들 상에 적용된 하나 이상의 제 2 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것으로서, 상기 하나 이상의 제 2 윈도우 파티션들은 상기 하나 이상의 제 1 윈도우 파티션들 중 인접한 것들 사이에 하나 이상의 경계들과 중첩되도록 시프트되는, 상기 셀프 어텐션을 결정하는 것; 및 상기 패치 병합 해제 엔진에 의해, 복수의 병합 해제된 패치들로 상기 하나 이상의 패치들의 각각의 패치를 세그먼트화하는 것으로서, 상기 복수의 병합 해제된 패치들은 중첩되지 않는, 상기 하나 이상의 패치들의 각각의 패치를 세그먼트화하는 것을 포함하는, 미디어 데이터를 프로세싱하는 방법.
- [0206] 양태 34: 양태 33에 있어서, 상기 복수의 디코더 트랜스포머 레이어들 중 제 2 디코더 트랜스포머 레이어의 제 1 트랜스포머 블록에 상기 복수의 병합 해제된 패치들을 제공하는 단계를 더 포함하는, 미디어 데이터를 프로세싱하는 방법.
- [0207] 양태 35: 양태 34에 있어서, 상기 제 2 디코더 트랜스포머 레이어의 패치 병합 해제 엔진에 의해, 상기 복수의 병합 해제된 패치들을 세그먼트화하는 단계; 및 상기 패치 병합 해제 엔진의 출력을 상기 복수의 디코더 트랜스포머 레이어들 중 제 3 디코더 트랜스포머 레이어에 제공하는 단계를 더 포함하는, 미디어 데이터를 프로세싱하는 방법.
- [0208] 양태 36: 양태 33 내지 35 중 어느 양태에서 있어서, 상기 복수의 병합 해제된 패치들 중 각각의 병합 해제된 패치는 균일한 패치 사이즈를 갖고 상기 패치 병합 해제 엔진은 2의 패치 사이즈 감소 팩터를 적용하는, 미디어 데이터를 프로세싱하는 방법.
- [0209] 양태 37: 양태 33 내지 36 중 어느 양태에서 있어서, 상기 복수의 병합 해제된 패치들로 상기 하나 이상의 패치들 중 각각의 패치를 세그먼트화하는 단계는 상기 복수의 병합 해제된 패치들의 특징 차원(feature dimension)을 감소시키는 것을 더 포함하는, 미디어 데이터를 프로세싱하는 방법.
- [0210] 양태 38: 양태 32 내지 37 중 어느 양태에서 있어서, 상기 복수의 디코더 트랜스포머 레이어들에 의해, 입력으로서 상기 인코딩된 이미지 데이터의 프레임의 잠재 표현을 수신하고 비선형 변환을 적용하여 디코딩된 이미지 데이터의 프레임을 생성하는 단계를 더 포함하는, 미디어 데이터를 프로세싱하는 방법.
- [0211] 양태 39: 양태 38에 있어서, 상기 비선형 변환은 합성 변환이고, 상기 디코딩된 이미지 데이터의 프레임은 상기 인코딩된 이미지 데이터의 프레임과 연관된 입력 이미지의 재구성인, 미디어 데이터를 프로세싱하는 방법.
- [0212] 양태 40: 양태 32 내지 39 중 어느 양태에서 있어서, 레이트-왜곡(rate-distortion)에 적어도 부분적으로 기초한 손실 함수를 사용하여 상기 복수의 디코더 트랜스포머 레이어들 중 하나 이상의 디코더 트랜스포머 레이어들을 훈련하는 단계를 더 포함하는, 미디어 데이터를 프로세싱하는 방법.
- [0213] 양태 41: 양태 40에 있어서, 상기 손실 함수는 레이트 왜곡을 위한 라그랑지 승수(Lagrangian multiplier)를 포함하는, 미디어 데이터를 프로세싱하는 방법.
- [0214] 양태 42: 양태 32 내지 41 중 어느 양태에서 있어서, 상기 적어도 하나의 디코더 트랜스포머 레이어에 포함된 상기 하나 이상의 트랜스포머 블록들 중 적어도 일부는 동일한 아키텍처를 갖는, 미디어 데이터를 프로세싱하는 방법.
- [0215] 양태 43: 양태 32 내지 42 중 어느 양태에서 있어서, 상기 적어도 하나의 디코더 트랜스포머 레이어에 포함된 상기 하나 이상의 트랜스포머 블록들의 각각은 동일한 아키텍처를 갖는, 미디어 데이터를 프로세싱하는 방법.
- [0216] 양태 44: 양태 32 내지 43 중 어느 양태에서 있어서, 상기 인코딩된 이미지 데이터의 프레임은 인코딩된 정지

이미지를 포함하는, 미디어 데이터를 프로세싱하는 방법.

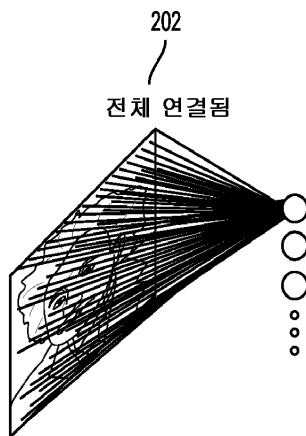
- [0217] 양태 45: 양태 32 내지 44 중 어느 양태에서 있어서, 상기 인코딩된 이미지 데이터의 프레임은 인코딩된 비디오 프레임을 포함하는, 미디어 데이터를 프로세싱하는 방법.
- [0218] 양태 46: 양태 32 내지 45 중 어느 양태에서 있어서, 더 포함하는, 적어도 제 1 훈련 데이터 세트 및 제 2 훈련 데이터 세트로 상기 복수의 디코더 트랜스포머 레이어들을 훈련하는 단계를 더 포함하고, 상기 제 2 훈련 데이터 세트의 데이터는 상기 제 1 훈련 데이터 세트의 데이터와 비교하여 역 시간 순서(reversed temporal order)를 갖는, 미디어 데이터를 프로세싱하는 방법.
- [0219] 양태 47: 양태 32 내지 46 중 어느 양태에서 있어서, 상기 복수의 디코더 트랜스포머 레이어들은 일련의 연속적인 디코더 트랜스포머 레이어들을 포함하는, 미디어 데이터를 프로세싱하는 방법.
- [0220] 양태 48: 미디어 데이터를 프로세싱하는 방법으로서, 프레임을 복수의 패치들로 세그먼트화하는 단계; 및 입력으로서 상기 복수의 패치들을 사용하여 인코더 서브네트워크의 복수의 인코더 트랜스포머 레이어들에 의해, 인코딩된 이미지 데이터의 프레임을 생성하는 단계를 포함하는, 미디어 데이터를 프로세싱하는 방법.
- [0221] 양태 49: 양태 48에 있어서, 상기 인코딩된 이미지 데이터의 프레임을 생성하는 단계는 상기 복수의 인코더 트랜스포머 레이어들 중 제 1 인코더 트랜스포머 레이어의 제 1 트랜스포머 블록에 의해, 하나 이상의 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것; 상기 제 1 인코더 트랜스포머 레이어의 제 2 트랜스포머 블록에 의해, 하나 이상의 시프트된 윈도우 파티션들 내에서 국부적으로 셀프 어텐션을 결정하는 것으로서, 상기 하나 이상의 시프트된 윈도우 파티션들은 상기 하나 이상의 윈도우 파티션들과 중첩하는, 상기 셀프 어텐션을 결정하는 것; 상기 제 1 트랜스포머 블록 및 상기 제 2 트랜스포머 블록 중 하나 이상에 의해, 상기 세그먼트화된 프레임에 비선형 변환을 적용하기 위한 특징들의 하나 이상의 패치들을 결정하는 것; 및 패치 병합 엔진에 의해, 상기 제 1 인코더 트랜스포머 레이어와 제 2 인코더 트랜스포머 레이어 사이의 패치 사이즈를 증가시키는 것을 포함하는, 미디어 데이터를 프로세싱하는 방법.
- [0222] 양태 50: 명령들이 저장된 비일시적 컴퓨터 판독가능 저장 매체로서, 상기 명령들은, 하나 이상의 프로세서들에 의해 실행될 때, 상기 하나 이상의 프로세서들로 하여금 양태 1 내지 16 및 양태 32 내지 47의 동작들 중 어느 것을 수행하게 하는, 비일시적 컴퓨터 판독가능 저장 매체.
- [0223] 양태 51: 양태 1 내지 16 및 양태 32 내지 47의 동작들 중 어느 것을 수행하기 위한 수단을 포함하는, 장치.
- [0224] 양태 52: 양태 17 내지 31의 동작들 중 어느 것을 수행하는 방법.
- [0225] 양태 53: 명령들이 저장된 비일시적 컴퓨터 판독가능 저장 매체로서, 상기 명령들은, 하나 이상의 프로세서들에 의해 실행될 경우, 상기 하나 이상의 프로세서들로 하여금 양태 17 내지 양태 31의 동작들 중 어느 것을 수행하게 하는, 비일시적 컴퓨터 판독가능 저장 매체.
- [0226] 양태 54: 양태들 17 내지 31의 동작들 중 어느 것을 수행하기 위한 수단을 포함하는, 장치.

도면

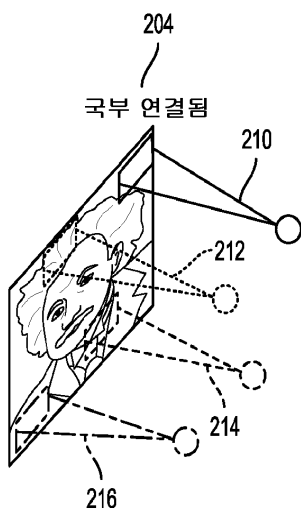
도면1



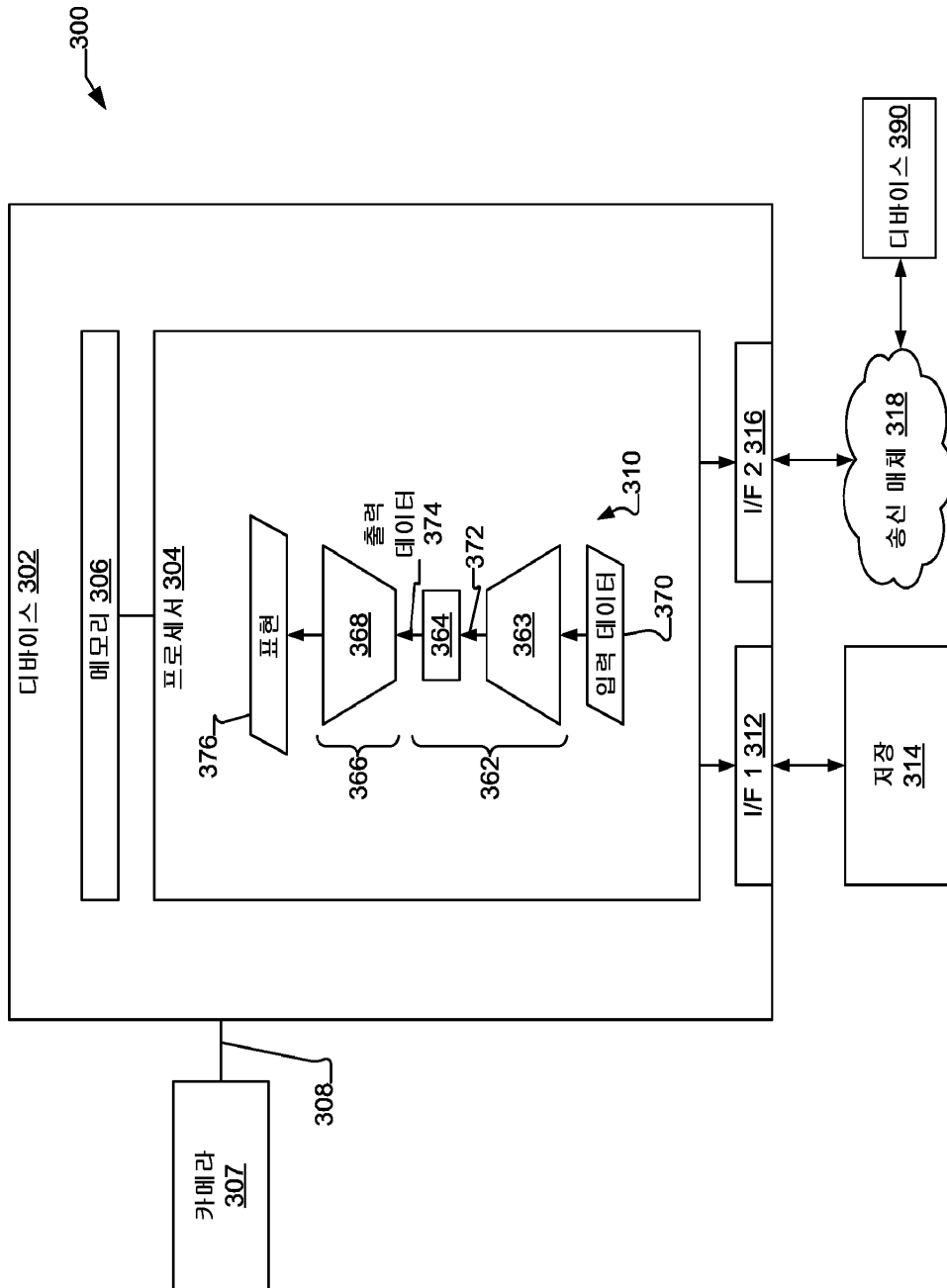
도면2a



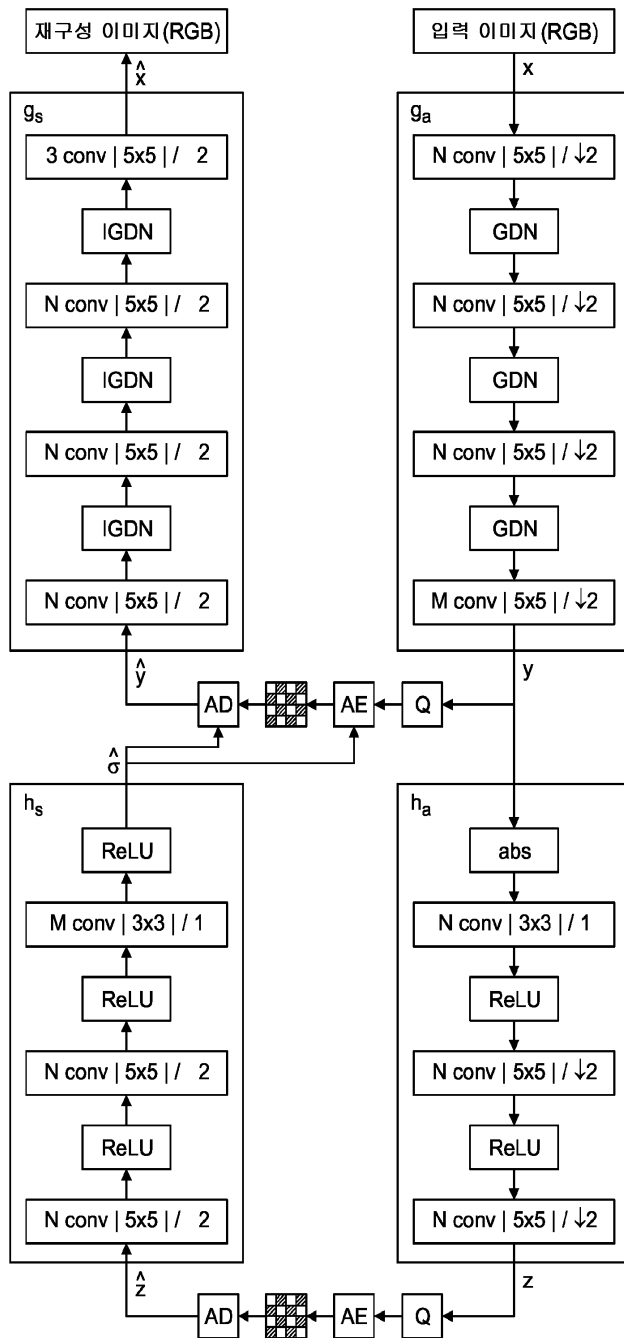
도면2b



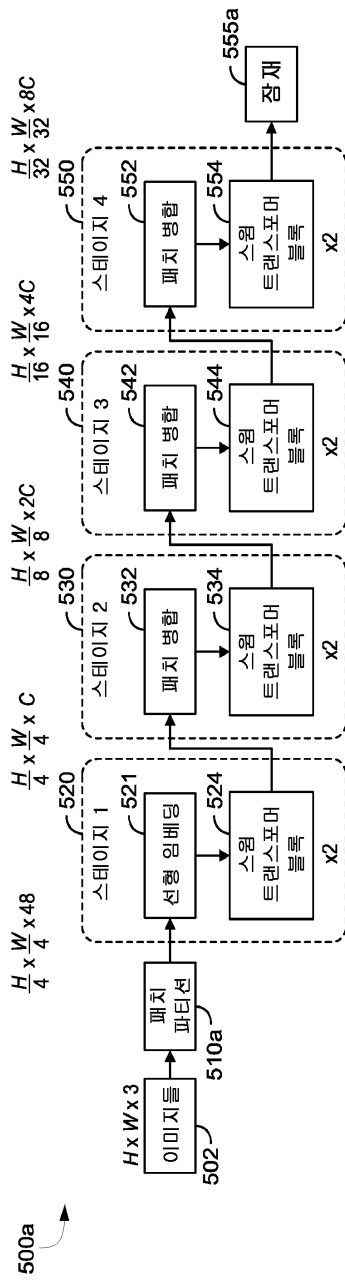
도면3



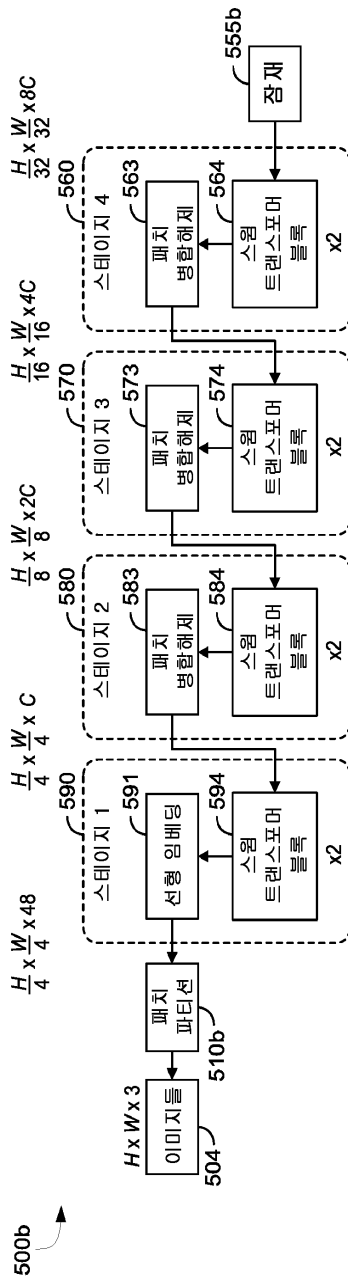
도면4



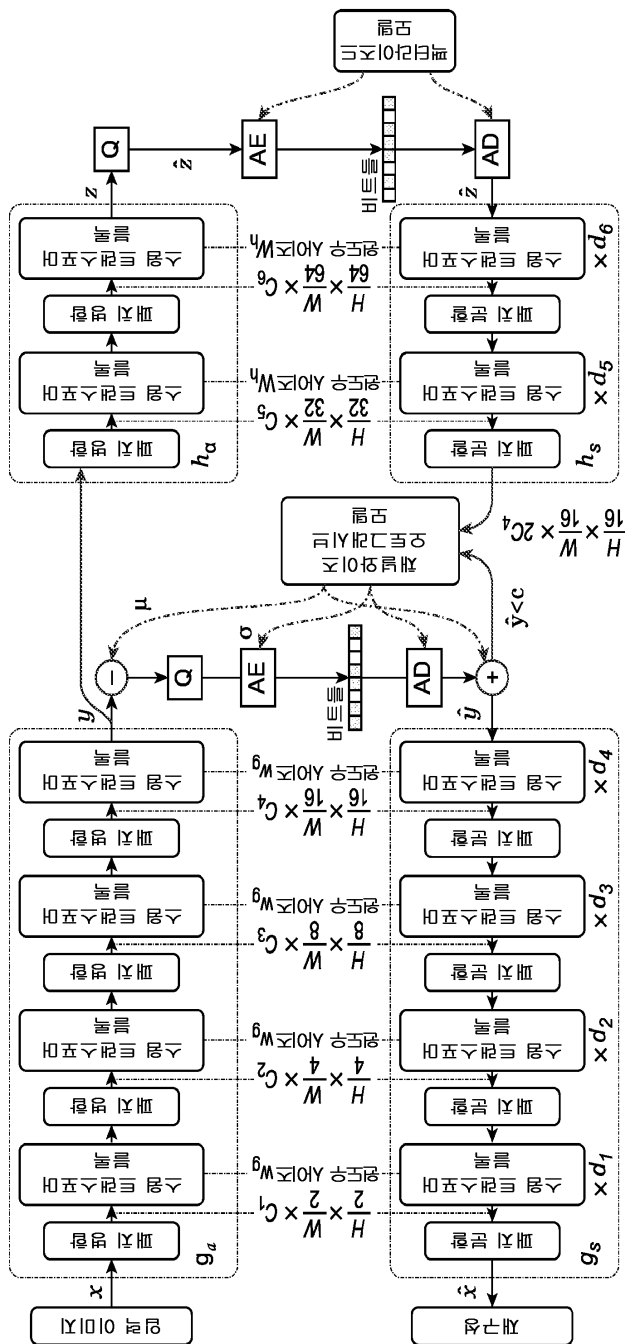
도면5a



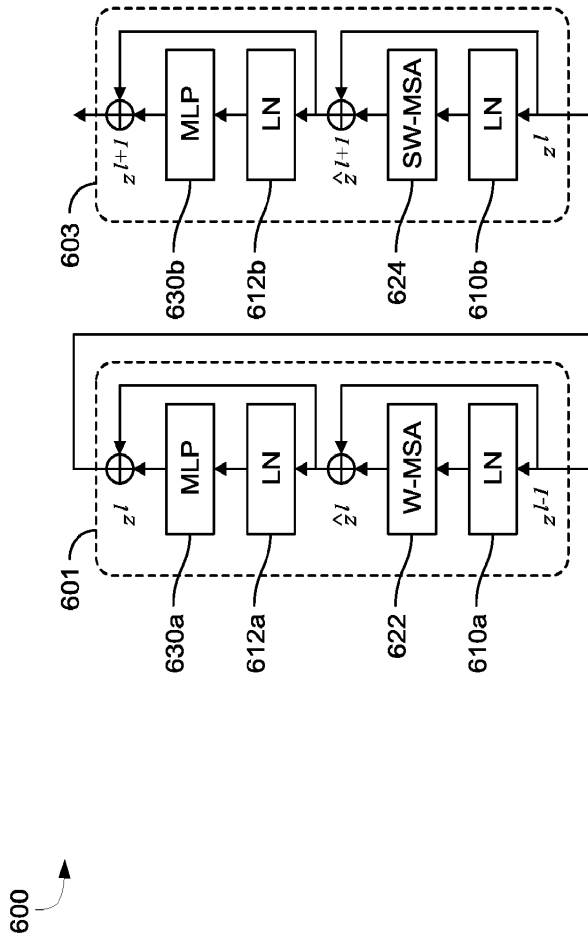
도면5b



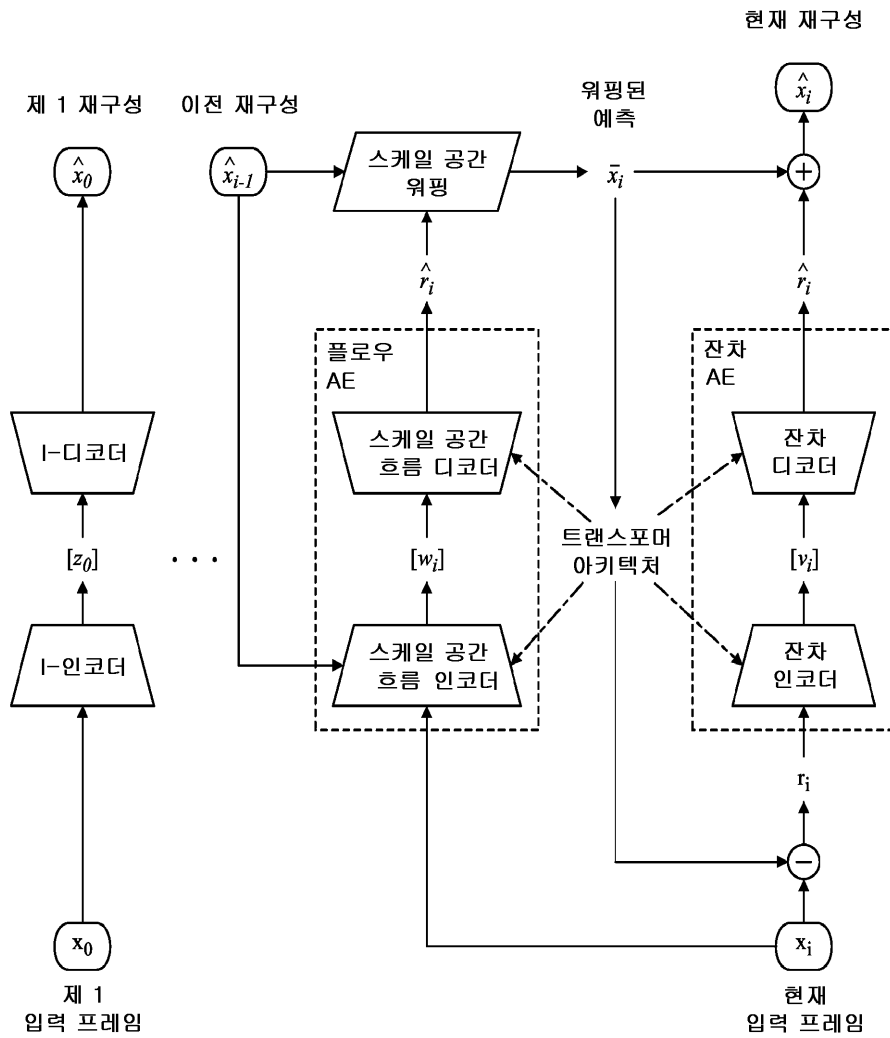
도면5c



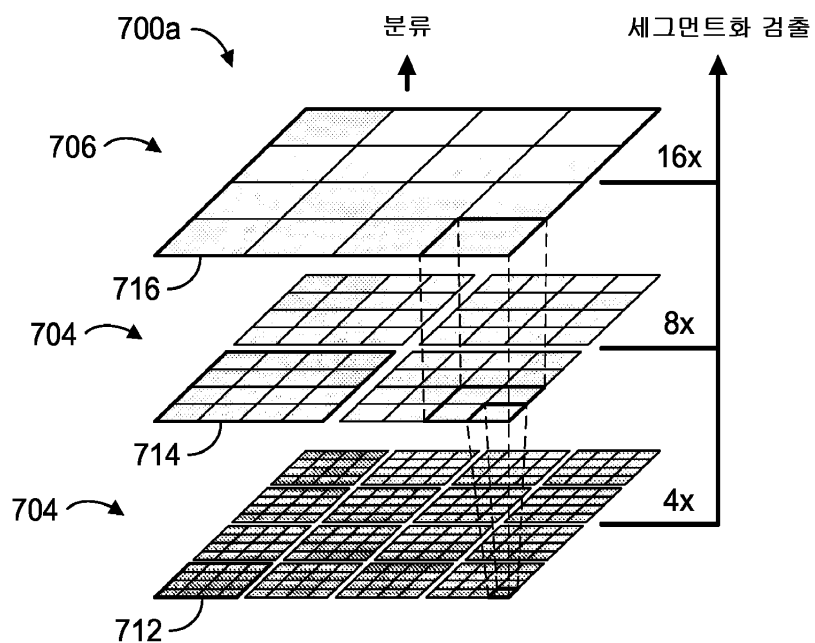
도면6a



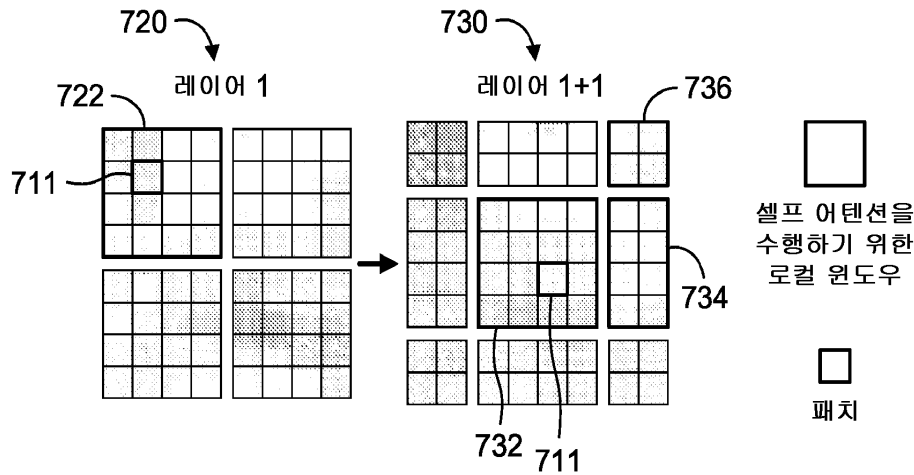
도면 6b



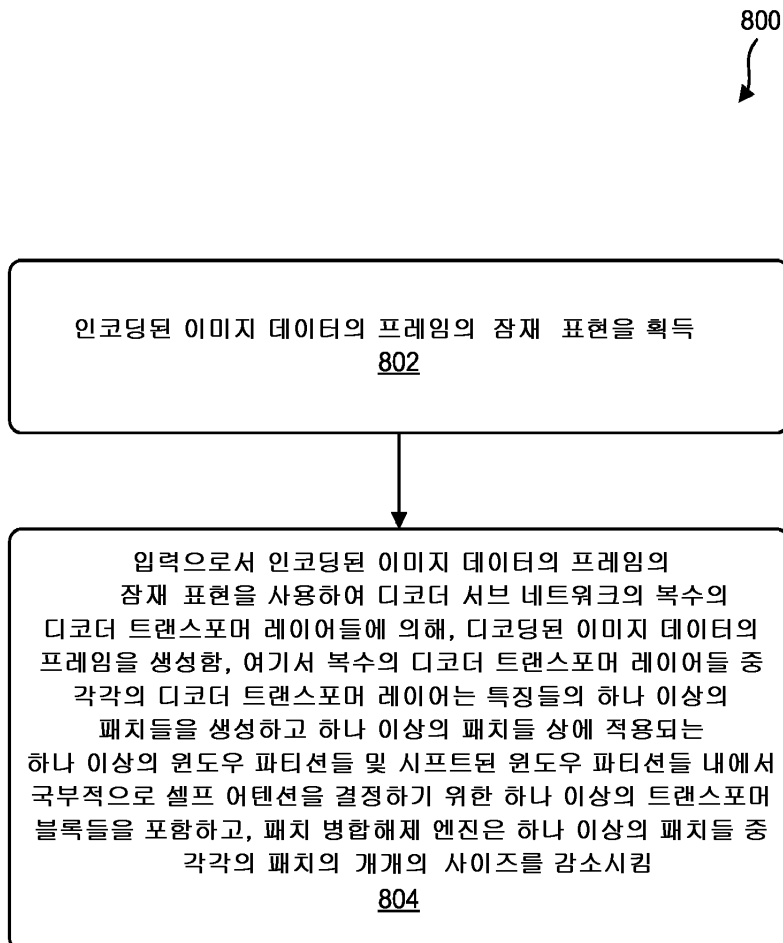
도면 7a



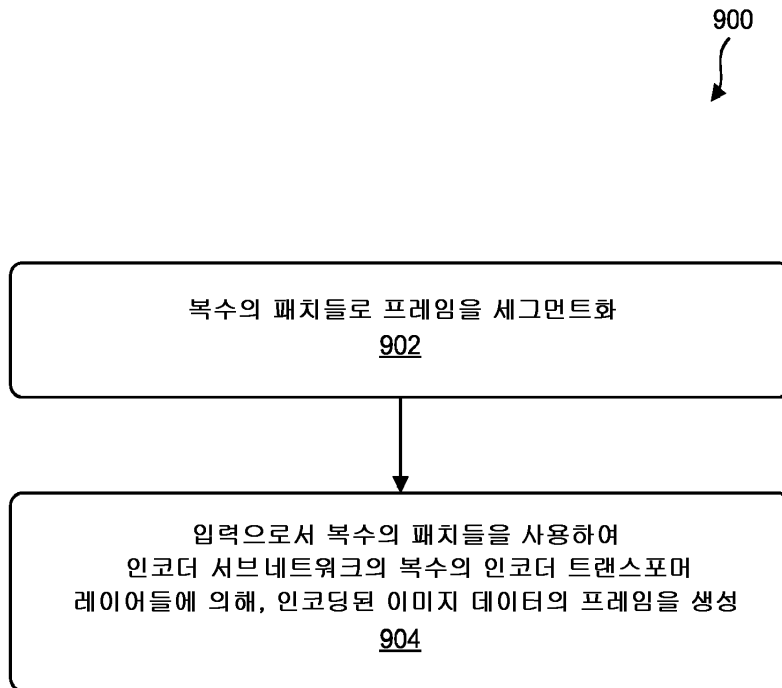
도면7b



도면8



도면9



도면10

