(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0150274
(43) 공개일자 2023년10월30일

(51) 국제특허분류(Int. Cl.)
H04N 19/537 (2014.01) *G06N 3/0464* (2023.01)
H04N 19/132 (2014.01) *H04N 19/159* (2014.01)
H04N 19/186 (2014.01) *H04N 19/52* (2014.01)
H04N 19/587 (2014.01)
(52) CPC특허분류
H04N 19/537 (2015.01)
G06N 3/0464 (2023.01)
(21) 출원번호 10-2023-7027621
(22) 출원일자(국제) 2022년02월22일
심사청구일자 없음
(85) 번역문제출일자 2023년08월14일
(86) 국제출원번호 PCT/US2022/017296
(87) 국제공개번호 WO 2022/182651
국제공개일자 2022년09월01일
(30) 우선권주장
63/153,475 2021년02월25일 미국(US)
17/676,510 2022년02월21일 미국(US)

(71) 출원인
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
싱 안키테쉬 쿠마르
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
에길메즈 힐미 에네스
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(뒷면에 계속)
(74) 대리인
특허법인코리아나

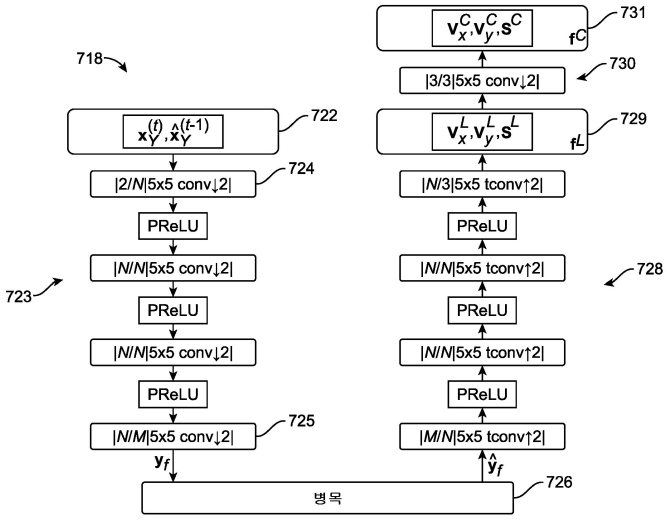
전체 청구항 수 : 총 30 항

(54) 발명의 명칭 **비디오 코딩을 위한 기계 학습 기반 플로우 결정**

(57) 요 약

시스템들 및 기법들은 비디오 데이터를 프로세싱하기 위하여 본원에 설명된다. 일부 양태들에서, 방법은 기계 학습 시스템에 의해, 입력 비디오 데이터를 획득하는 단계를 포함할 수 있다. 입력 비디오 데이터는 현재 프레임에 대한 하나 이상의 휘도 성분들을 포함한다. 방법은 기계 학습 시스템에 의해, 현재 프레임에 대한 휘도 성분(들)을 사용하여 현재 프레임의 휘도 성분(들)에 대한 모션 정보 및 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하는 단계를 포함할 수 있다. 일부 경우들에서, 방법은 현재 프레임의 루마 성분(들) 및 이전 프레임의 적어도 하나의 재구성된 루마 성분에 기초하여 휘도 성분(들)에 대한 모션 정보를 결정하는 단계를 포함할 수 있다. 일부 경우들에서, 방법은 현재 프레임의 휘도 성분(들)에 대해 결정된 모션 정보를 사용하여 현재 프레임의 색차 성분(들)에 대한 모션 정보를 결정하는 단계를 더 포함할 수 있다.

대 표 도



(52) CPC특허분류
H04N 19/132 (2015.01)
H04N 19/159 (2015.01)
H04N 19/186 (2015.01)
H04N 19/52 (2015.01)
H04N 19/587 (2015.01)
(72) 발명자
코반 무하메드 제이드
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
카르체비츠 마르타
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775

명 세 서

청구범위

청구항 1

비디오 데이터를 프로세싱하는 방법으로서,

기계 학습 시스템에 의해, 현재 프레임에 대한 적어도 하나의 휘도 성분을 포함하는 입력 비디오 데이터를 획득하는 단계; 및

상기 기계 학습 시스템에 의해, 상기 현재 프레임에 대한 상기 적어도 하나의 휘도 성분을 사용하여, 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 모션 정보 및 상기 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하는 단계를 포함하는, 방법.

청구항 2

제 1 항에 있어서, 추가로

상기 기계 학습 시스템에 의해, 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 모션 정보 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 모션 정보를 사용하여, 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 워핑 파라미터 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 하나 이상의 워핑 파라미터들을 결정하는 단계; 및

상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 상기 워핑 파라미터 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 하나 이상의 워핑 파라미터들을 사용하여, 상기 현재 프레임에 대한 하나 이상의 인터-프레임 예측들을 결정하는 단계를 포함하는, 방법.

청구항 3

제 2 항에 있어서, 상기 하나 이상의 인터-프레임 예측들은 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 상기 워핑 파라미터 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 하나 이상의 워핑 파라미터들을 사용하여 보간 연산 (interpolation operation) 을 적용함으로써 적어도 부분적으로 결정되는, 방법.

청구항 4

제 3 항에 있어서, 상기 보간 연산은 삼선형 보간 (trilinear interpolation) 연산을 포함하는, 방법.

청구항 5

제 2 항에 있어서, 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 상기 워핑 파라미터 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 하나 이상의 워핑 파라미터들은 공간-스케일 플로우 (SSF) 워핑 파라미터들을 포함하는, 방법.

청구항 6

제 5 항에 있어서, 상기 SSF 워핑 파라미터들은 학습된 스케일-플로우 벡터들을 포함하는, 방법.

청구항 7

제 1 항에 있어서, 상기 현재 프레임에 대한 상기 적어도 하나의 휘도 성분을 사용하여 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 상기 모션 정보 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 모션 정보를 결정하는 단계는,

상기 현재 프레임의 상기 적어도 하나의 휘도 성분 및 이전 프레임의 적어도 하나의 재구성된 루마 성분에 기초하여 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 상기 모션 정보를 결정하는 단계; 및

상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대해 결정된 상기 모션 정보를 사용하여 상기 현재 프레임

의 상기 하나 이상의 색차 성분들에 대한 상기 모션 정보를 결정하는 단계를 포함하는, 방법.

청구항 8

제 7 항에 있어서, 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 모션 정보는 상기 기계 학습 시스템의 컨볼루션 계층 (convolutional layer) 을 사용하여 결정되는, 방법.

청구항 9

제 7 항에 있어서, 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 모션 정보는 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대해 결정된 상기 모션 정보를 샘플링함으로써 적어도 부분적으로 결정되는, 방법.

청구항 10

제 1 항에 있어서, 상기 현재 프레임은 비디오 프레임을 포함하는, 방법.

청구항 11

제 1 항에 있어서, 상기 하나 이상의 색차 성분들은 적어도 하나의 색차-청색 성분 및 색차-적색 성분을 포함하는, 방법.

청구항 12

제 1 항에 있어서, 상기 현재 프레임은 휘도-색차 (YUV) 포맷을 갖는, 방법.

청구항 13

제 12 항에 있어서, 상기 YUV 포맷은 YUV 4:2:0 포맷인, 방법.

청구항 14

비디오 데이터를 프로세싱하기 위한 장치로서,

적어도 하나의 메모리; 및

상기 적어도 하나의 메모리에 커플링된 하나 이상의 프로세서들을 포함하고, 상기 하나 이상의 프로세서들은

기계 학습 시스템을 사용하여, 현재 프레임에 대한 적어도 하나의 휘도 성분을 포함하는 입력 비디오 데이터를 획득하고, 그리고

상기 기계 학습 시스템을 사용하여, 상기 현재 프레임에 대한 상기 적어도 하나의 휘도 성분을 사용하여, 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 모션 정보 및 상기 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하도록 구성되는, 장치.

청구항 15

제 14 항에 있어서, 상기 하나 이상의 프로세서들은,

상기 기계 학습 시스템을 사용하여, 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 모션 정보 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 모션 정보에 기초하여, 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 워핑 파라미터 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 하나 이상의 워핑 파라미터들을 결정하고, 그리고

상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 상기 워핑 파라미터 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 하나 이상의 워핑 파라미터들을 사용하여, 상기 현재 프레임에 대한 하나 이상의 인터-프레임 예측들을 결정하도록 구성되는, 장치.

청구항 16

제 15 항에 있어서, 상기 하나 이상의 인터-프레임 예측들은 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 상기 워핑 파라미터 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 하나 이상의 워

핑 파라미터들을 사용하여 보간 연산을 적용함으로써 적어도 부분적으로 결정되는, 장치.

청구항 17

제 16 항에 있어서, 상기 보간 연산은 삼선형 보간 연산을 포함하는, 장치.

청구항 18

제 15 항에 있어서, 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 상기 워핑 파라미터 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 하나 이상의 워핑 파라미터들은 공간-스케일 플로우 (SSF) 워핑 파라미터들을 포함하는, 장치.

청구항 19

제 18 항에 있어서, 상기 SSF 워핑 파라미터들은 학습된 스케일-플로우 벡터들을 포함하는, 장치.

청구항 20

제 14 항에 있어서, 상기 현재 프레임에 대한 상기 적어도 하나의 휘도 성분을 사용하여 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 상기 모션 정보 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 모션 정보를 결정하기 위해, 상기 하나 이상의 프로세서들은,

상기 현재 프레임의 상기 적어도 하나의 휘도 성분 및 이전 프레임의 적어도 하나의 재구성된 루마 성분에 기초하여 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 상기 모션 정보를 결정하고, 그리고

상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대해 결정된 상기 모션 정보를 사용하여 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 모션 정보를 결정하도록 구성되는, 장치.

청구항 21

제 20 항에 있어서, 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 모션 정보는 상기 기계 학습 시스템의 컨볼루션 계층을 사용하여 결정되는, 장치.

청구항 22

제 20 항에 있어서, 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 모션 정보를 결정하기 위해, 상기 하나 이상의 프로세서들은 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대해 결정된 상기 모션 정보를 샘플링하도록 구성되는, 장치.

청구항 23

제 14 항에 있어서, 상기 현재 프레임은 비디오 프레임을 포함하는, 장치.

청구항 24

제 14 항에 있어서, 상기 하나 이상의 색차 성분들은 적어도 하나의 색차-청색 성분 및 색차-적색 성분을 포함하는, 장치.

청구항 25

제 14 항에 있어서, 상기 현재 프레임은 휘도-색차 (YUV) 포맷을 갖는, 장치.

청구항 26

제 25 항에 있어서, 상기 YUV 포맷은 YUV 4:2:0 포맷인, 장치.

청구항 27

제 14 항에 있어서, 하나 이상의 프레임들을 캡처하도록 구성된 적어도 하나의 카메라를 더 포함하는, 장치.

청구항 28

제 14 항에 있어서, 하나 이상의 프레임들을 디스플레이하도록 구성된 적어도 하나의 디스플레이를 더 포함하는, 장치.

청구항 29

제 14 항에 있어서, 상기 장치는 모바일 디바이스를 포함하는, 장치.

청구항 30

명령들이 저장된 비일시적 컴퓨터 판독가능 저장 매체로서, 상기 명령들은, 하나 이상의 프로세서들에 의해 실행될 경우, 상기 하나 이상의 프로세서들로 하여금

기계 학습 시스템을 사용하여, 현재 프레임에 대한 적어도 하나의 휘도 성분을 포함하는 입력 비디오 데이터를 획득하게 하고, 그리고

상기 기계 학습 시스템을 사용하여, 상기 현재 프레임에 대한 상기 적어도 하나의 휘도 성분을 사용하여, 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 모션 정보 및 상기 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하게 하는, 비일시적 컴퓨터 판독가능 저장 매체.

발명의 설명

기 술 분 야

[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 하나 이상의 이미지 프레임들 또는 픽처들 (예를 들어, 비디오 프레임들/픽처들) 의 루마 및 크로마 성분들에 대한 플로우 정보를 결정하기 위한 기법들에 관한 것이다.

배 경 기 술

[0002] 많은 디바이스들 및 시스템들은 비디오 데이터가 프로세싱되고 소비를 위해 출력될 수 있게 한다. 디지털 비디오 데이터는 소비자들 및 비디오 제공자들의 수요들을 충족시키기 위해 대량의 데이터를 포함한다. 예를 들어, 비디오 데이터의 소비자들은 높은 충실도, 해상도, 프레임 레이트들 등을 갖는 높은 품질의 비디오를 원한다. 결과적으로, 이들 요구들을 충족시키기 위해 필요한 대량의 비디오 데이터는 그 비디오 데이터를 프로세싱하고 저장하는 통신 네트워크들 및 디바이스들에 부담을 지운다.

[0003] 비디오 데이터를 압축하기 위해 다양한 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. 끊임없이 진화하는 비디오 서비스들이 이용가능하게 됨에 따라, 우수한 코딩 효율을 갖는 인코딩 기법들이 필요하다.

발명의 내용

[0004] 하나 이상의 머신 학습 시스템들을 사용하여 이미지 및/또는 비디오 콘텐츠를 코딩(예를 들어, 인코딩 및/또는 디코딩)하기 위한 시스템들 및 기술들이 설명된다. 적어도 하나의 예에 따르면, 비디오를 프로세싱하기 위한 방법이 제공된다. 이 방법은, 기계 학습 시스템에 의해, 입력 비디오 데이터를 획득하는 단계 - 입력 비디오 데이터는 현재 프레임에 대한 적어도 하나의 휘도 성분을 포함함 -; 및 기계 학습 시스템에 의해, 현재 프레임에 대한 적어도 하나의 휘도 성분을 사용하여 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보 및 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하는 단계를 포함한다.

[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, 회로부로 구현됨) 을 포함하는, 비디오 데이터를 프로세싱하기 위한 장치가 제공된다. 하나 이상의 프로세서들은, 기계 학습 시스템을 사용하여, 입력 비디오 데이터를 획득하도록 구성되고, 입력 비디오 데이터는 현재 프레임에 대한 적어도 하나의 휘도 성분을 포함하고; 그리고 기계 학습 시스템을 사용하여, 현재 프레임에 대한 적어도 하나의 휘도 성분을 사용하여 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보 및 현재 프레임의 하나 이상의 색차 컴포넌트들에 대한 모션 정보를 결정하도록 구성되고 그러한 것을 수행할 수 있다.

[0006] 다른 예에서, 하나 이상의 프로세서들에 의해 실행될 때, 하나 이상의 프로세서들로 하여금: 기계 학습 시스템을 사용하여, 입력 비디오 데이터를 획득하게 하고 - 입력 비디오 데이터는 현재 프레임에 대한 적어도 하나의 휘도 성분을 포함함 -; 기계 학습 시스템을 사용하여, 현재 프레임에 대한 적어도 하나의 휘도 성분을 사용하여 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보 및 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하게 하는 명령들을 저장한 비일시적 컴퓨터 판독가능 매체가 제공된다.

[0007] 다른 예에 있어서, 비디오 데이터를 프로세싱하기 위한 장치가 제공된다. 이 장치는, 입력 비디오 데이터를 획득하는 수단으로서, 입력 비디오 데이터는 현재 프레임에 대한 적어도 하나의 휘도 성분을 포함하는, 상기 입력 비디오 데이터를 획득하는 수단; 및 현재 프레임에 대한 적어도 하나의 휘도 성분을 이용하여, 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보 및 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하는 수단을 포함한다.

[0008] 일부 양태들에서, 앞서 설명된 방법들, 장치들, 및 컴퓨터 판독가능 매체 중 하나 이상은, 기계 학습 시스템에 의해, 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보 및 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 사용하여, 현재 프레임의 적어도 하나의 휘도 성분에 대한 워핑 파라미터 및 현재 프레임의 하나 이상의 색차 성분들에 대한 하나 이상의 워핑 파라미터들을 결정하는 것; 및 현재 프레임의 적어도 하나의 휘도 성분에 대한 워핑 파라미터 및 현재 프레임의 하나 이상의 색차 성분들에 대한 하나 이상의 워핑 파라미터들을 사용하여 현재 프레임에 대한 하나 이상의 인터-프레임 예측들을 결정하는 것을 더 포함한다.

[0009] 일부 양태들에서, 상기 하나 이상의 인터-프레임 예측들은 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 상기 워핑 파라미터 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 하나 이상의 워핑 파라미터들을 사용하여 보간 연산을 적용함으로써 적어도 부분적으로 결정된다.

[0010] 일부 양태들에서, 보간 연산은 삼선형 보간 연산을 포함한다.

[0011] 일부 양태들에서, 현재 프레임의 적어도 하나의 휘도 성분에 대한 워핑 파라미터 및 현재 프레임의 하나 이상의 색차 성분들에 대한 하나 이상의 워핑 파라미터들은 공간-스케일 플로우 (SSF) 워핑 파라미터들을 포함한다.

[0012] 일부 양태들에서, SSF 워핑 파라미터들은 학습된 스케일-플로우 벡터들을 포함한다.

[0013] 일부 양태들에서, 현재 프레임에 대한 적어도 하나의 휘도 성분을 사용하여 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보 및 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하기 위해, 앞서 설명된 방법들, 장치들, 및 컴퓨터 판독가능 매체 중 하나 이상은, 현재 프레임의 적어도 하나의 휘도 성분 및 이전 프레임의 적어도 하나의 재구성된 루마 성분에 기초하여 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보를 결정하는 것; 및 현재 프레임의 적어도 하나의 휘도 성분에 대해 결정된 모션 정보를 사용하여 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하는 것을 더 포함한다.

[0014] 일부 양태들에서, 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보는 기계 학습 시스템의 컨볼루션 계층을 사용하여 결정된다.

[0015] 일부 양태들에서, 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하기 위해, 앞서 설명된 방법들, 장치들, 및 컴퓨터 판독가능 매체 중 하나 이상은 현재 프레임의 적어도 하나의 휘도 성분에 대해 결정된 모션 정보를 샘플링하는 것을 더 포함한다.

[0016] 일부 양태들에서, 현재 프레임은 비디오 프레임을 포함한다.

[0017] 일부 양태들에서, 하나 이상의 색차 성분들은 적어도 하나의 색차-청색 성분 및 적어도 하나의 색차-적색 성분을 포함한다.

[0018] 일부 양태들에서, 현재 프레임은 휘도-색차 (YUV) 포맷을 갖는다. 일부 경우에, YUV 포맷은 YUV 4:2:0 포맷이다.

[0019] 일부 양태들에서, 본 명세서에 설명된 장치들은 모바일 디바이스 (예를 들어, 모바일 전화 또는 소위 "스마트 폰", 태블릿 컴퓨터, 또는 다른 유형의 모바일 디바이스), 웨어러블 디바이스, 확장 현실 디바이스 (예를 들어, 가상 현실 (VR) 디바이스, 증강 현실 (AR) 디바이스, 또는 혼합 현실 (MR) 디바이스), 개인용 컴퓨터, 랩톱 컴퓨터, 비디오 서버, 텔레비전, 차량 (또는 차량의 컴퓨팅 디바이스), 또는 다른 디바이스를 포함하거나 그 일부일 수 있다. 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 신택스 데이터를 송신 매체를 통해 적어도 하나의 디바이스로 송신하도록 구성된 송신기를 포함한다. 일부 양태들에서, 프로세서는 뉴럴 프로세싱 유닛 (NPU), 중앙 프로세싱 유닛 (CPU), 그래픽 프로세싱 유닛 (GPU), 또는 다른 프로세싱 디바이스 또는 컴포넌트를 포함한다.

[0020] 이 개요는, 청구된 요지의 핵심적인 또는 본질적인 특징들을 식별하도록 의도되지 않으며, 청구된 요지의 범위를 결정하는 데 별개로 사용되도록 의도되지도 않는다. 그 주제는 이 특허의 전체 명세서, 임의의 또는 모든 도면들, 및 각각의 청구항의 적절한 부분들을 참조하여 이해되어야 한다.

[0021] 전술한 내용은, 다른 특징 및 실시 형태들과 함께, 다음의 명세서, 청구항 및 첨부 도면을 참조하면 더욱 명백해질 것이다.

도면의 간단한 설명

[0022] 본 출원의 예시적 실시형태들은 다음 도면들을 참조하여 이하에서 상세히 설명된다.

도 1 은 시스템-온-칩(SOC)의 예시적인 구현을 도시한다.

도 2a 는 완전히 연결된 뉴럴 네트워크의 예를 도시한다.

도 2b 는 국부적으로 연결된 뉴럴 네트워크의 예를 도시한다.

도 2c 는 컨볼루션 뉴럴 네트워크의 예를 도시한다.

도 2d 는 이미지로부터 시각적 피처들을 인식하도록 설계된 딥 컨볼루션 네트워크(DCN)의 상세한 예를 예시한다.

도 3 은 딥 컨볼루션 네트워크(DCN)를 설명하기 위한 블록도이다.

도 4 는 일부 예들에 따른, 뉴럴 네트워크 기반 시스템을 사용하여 이미지 및/또는 비디오 코딩 (인코딩 및 디코딩) 을 수행하도록 동작가능한 디바이스를 포함하는 시스템의 일 예를 예시하는 다이어그램이다.

도 5 는 일부 예들에 따른, 적색-녹색-청색(RGB) 포맷을 갖는 입력을 위한 엔드-투-엔드 뉴럴 네트워크 기반 이미지 및 비디오 코딩 시스템의 예를 예시하는 다이어그램이다.

도 6 은 일부 예들에 따른, 엔드-투-엔드 뉴럴 네트워크 기반 이미지 및 비디오 코딩 시스템의 일부일 수 있는 하나 이상의 휘도-색차(YUV) 입력 포맷들(예를 들어, 4:2:0 YUV 입력 포맷)을 프로세싱하도록 구성된 공간-스케일 흐름(SSF) 뉴럴 네트워크 아키텍처의 예를 예시하는 다이어그램이다.

도 7a 는 일부 예들에 따른, 루마 입력으로 동작하는 기계-학습 기반 플로우 엔진의 예를 예시하는 다이어그램이다.

도 7b 는 일부 예들에 따른, 크로마 모션 정보를 획득하기 위한 루마 모션 정보의 서브샘플링의 예를 예시하는 다이어그램이다.

도 8a 는 일부 예들에 따른, YUV(예를 들어, YUV 4:2:0) 잔차들을 갖는 기계 학습 기반 아키텍처의 예를 예시하는 다이어그램이다.

도 8b 는 일부 예들에 따른, 1x1 컨볼루션 계층의 예시적인 동작을 예시하는 다이어그램이다.

도 9 는 일부 예들에 따른, YUV 4:2:0 입력과 같은 YUV 입력(Y, U, 및 V)과 직접 작동하는 (예를 들어, 엔드-투-엔드 뉴럴 네트워크 기반 이미지 및 비디오 코딩 시스템의) 기계 학습 기반 아키텍처의 예를 예시하는 다이어그램이다.

도 10 은 일부 예들에 따른, YUV 4:2:0 입력과 같은 YUV 입력(Y, U, 및 V)과 직접 작동하는 (예를 들어, 엔드-투-엔드 뉴럴 네트워크 기반 이미지 및 비디오 코딩 시스템의) 기계 학습 기반 아키텍처의 또 다른 예를 예시하는 다이어그램이다.

도 11 은 일부 예들에 따른, 비디오 데이터를 프로세싱하기 위한 프로세스의 일례를 나타내는 흐름도이다.

도 12 는 본 명세서에 설명된 다양한 기술을 구현할 수 있는 예시적인 컴퓨팅 디바이스의 예시적인 컴퓨팅 디바

이스 아키텍처를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0023] 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 목적들로, 특정 상세들이 본 출원의 실시형태들의 철저한 이해를 제공하기 위해 제시된다. 하지만, 다양한 실시형태들이 이들 특정 상세들 없이 실시될 수도 있음이 명백할 것이다. 도면들 및 설명은 제한적인 것으로 의도되지 않는다.

[0024] 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 청구범위에 설명된 바와 같이 본 출원의 사상 및 범위를 벗어나지 않으면서 엘리먼트들의 기능 및 배열에 다양한 변경들이 이루어질 수도 있음이 이해되어야 한다.

[0025] 디지털 비디오 데이터는, 특히 고품질 비디오 데이터에 대한 요구가 계속 증가함에 따라, 많은 양의 데이터를 포함할 수 있다. 예를 들어, 비디오 데이터의 소비자들은 통상적으로, 높은 충실도, 해상도, 프레임 레이트들 등을 갖는 점점 더 높은 품질의 비디오를 원한다. 그러나, 이러한 요구를 만족시키기 위해 요구되는 많은 양의 비디오 데이터는 비디오 데이터를 처리하고 저장하는 장치들뿐만 아니라 통신 네트워크에도 상당한 부담을 줄 수 있다.

[0026] 비디오 데이터를 코딩하기 위해 다양한 기법들이 사용될 수 있다. 비디오 코딩은 특정 비디오 코딩 표준에 따라 수행될 수 있다. 예시적인 비디오 코딩 표준들은 고효율 비디오 코딩 (HEVC), 진보된 비디오 코딩 (AVC), 동영상 전문가 그룹 (MPEG) 코딩, 및 다목적 비디오 코딩 (VVC) 을 포함한다. 비디오 코딩은 종종 인터-예측(inter-prediction) 또는 인트라-예측(intra-prediction)과 같은 예측 방법들을 사용하며, 이는 비디오 이미지들 또는 시퀀스들에 존재하는 리던던시들을 이용한다. 비디오 코딩 기법들의 공통 목적은 비디오 품질에 대한 열화들을 회피 또는 최소화하면서 더 낮은 비트 레이트를 사용하는 형태로 비디오 데이터를 압축하는 것이다. 비디오 서비스들에 대한 수요가 증가하고 새로운 비디오 서비스들이 이용가능해짐에 따라, 더 양호한 코딩 효율, 성능, 및 레이트 제어를 갖는 코딩 기법들이 필요하다.

[0027] 기계 학습(ML) 기반 시스템들은 이미지 및/또는 비디오 코딩을 수행하는 데 사용될 수 있다. 일반적으로 ML은 인공 지능(AI)의 부분 집합이다. ML 시스템들은, 컴퓨터 시스템들이 명시적 명령어들의 사용 없이, 패턴들 및 추론에 의존함으로써 다양한 태스크들을 수행하기 위해 사용할 수 있는 알고리즘들 및 통계 모델들을 포함할 수 있다. ML 시스템의 일 예는 인공 뉴런들(예를 들어, 뉴런 모델들)의 상호연결된 그룹을 포함할 수 있는 뉴럴 네트워크(인공 뉴럴 네트워크로도 지칭됨)이다. 뉴럴 네트워크들은 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.

[0028] 뉴럴 네트워크 내의 개별 노드들은 입력 데이터를 취하고 데이터에 대해 간단한 연산들을 수행함으로써 생물학적 뉴런들을 에뮬레이트할 수 있다. 입력 데이터에 대해 수행된 단순한 동작들의 결과들은 다른 뉴런들에 선택적으로 전달된다. 가중치 값들은 네트워크 내의 각각의 벡터 및 노드와 연관되고, 이들 값들은 입력 데이터가 출력 데이터와 어떻게 관련되는지를 제약한다. 예를 들어, 각 노드의 입력 데이터는 대응하는 가중치 값과 곱해질 수 있다. 곱들의 합은 선택적인 바이어스에 의해 조정될 수 있고, 활성화 함수가 결과에 적용되어, 노드의 출력 신호 또는 "출력 활성화"(때때로 활성화 맵 또는 특징 맵으로 지칭됨)를 산출할 수 있다. 가중치 값들은 초기에 네트워크를 통한 트레이닝 데이터의 반복 흐름에 의해 결정될 수 있다(예를 들어, 가중치 값들은 네트워크가 그들의 전형적인 입력 데이터 특성들에 의해 특정 클래스들을 식별하는 방법을 학습하는 트레이닝 단계 동안 설정된다).

[0029] 특히, 컨볼루션 뉴럴 네트워크(CNN), 순환 뉴럴 네트워크(RNN), 생성 적대 네트워크(GAN), 다층 퍼셉트론(MLP) 뉴럴 네트워크와 같은 상이한 유형의 뉴럴 네트워크가 존재한다. 예를 들어, 컨볼루션 뉴럴 네트워크(CNN)는 피드-포워드 인공 뉴럴 네트워크의 일종이다. 컨볼루션 뉴럴 네트워크들은 각각 수용 필드(receptive field)(예를 들어, 입력 공간의 공간적으로 로컬화된 영역)를 갖고 입력 공간을 집합적으로 타일링하는 인공 뉴런들의 집합들을 포함할 수 있다. RNN들은 계층의 결과를 예측하는 것을 돕기 위해 계층의 출력을 저장하고 이 출력을 입력에 다시 공급하는 원리에 대해 작동한다. GAN은 입력 데이터에서 패턴들을 학습할 수 있는 생성 뉴럴 네트워크의 형태이며, 따라서 뉴럴 네트워크 모델은 원래 데이터세트로부터 합리적으로 얻어질 수 있었던 새로운 합

성 출력들을 생성할 수 있다. GAN은 합성된 출력을 생성하는 생성 뉴럴 네트워크 및 진본성에 대해 출력을 평가하는 판별 뉴럴 네트워크를 포함하여 함께 동작하는 2개의 뉴럴 네트워크를 포함할 수 있다. MLP 뉴럴 네트워크들에서, 데이터는 입력 계층에 공급될 수 있고, 하나 이상의 은닉 계층들은 데이터에 대한 추상화의 레벨들을 제공한다. 그런 다음, 추상화된 데이터에 기초하여 출력 계층 상에서 예측들이 이루어질 수 있다.

[0030] 계층화된 뉴럴 네트워크 아키텍처들(다수의 은닉 층들이 존재할 때 심층 뉴럴 네트워크들로 지칭됨)에서, 인공 뉴런들의 제 1 층의 출력은 인공 뉴런들의 제 2 층에 대한 입력이 되고, 인공 뉴런들의 제 2 층의 출력은 인공 뉴런들의 제 3 층에 대한 입력이 되고, 등등이다. CNN들은, 예를 들어, 특징들의 계층을 인식하도록 훈련될 수 있다. CNN 아키텍처들에서의 계산은 하나 이상의 계산 체인으로 구성될 수 있는 프로세싱 노드들의 집단에 걸쳐 분산될 수 있다. 이들 멀티-계층화된 아키텍처들은 한번에 하나의 계층씩 트레이닝될 수도 있고, 역 전파(back propagation) 를 이용하여 미세-튜닝될 수도 있다.

[0031] 많은 경우들에서, 딥 러닝 기반 시스템들은 엔트로피 코딩에 사용되는 양자화된 레이턴시들에 걸쳐 확률 모델을 학습하는 것을 담당하는 오토인코더 서브-네트워크 (인코더 서브-네트워크) 및 제 2 서브-네트워크 (일부 경우들에서 하이퍼프라이어 (hyperprior) 네트워크라고도 지칭됨) 의 조합으로서 설계된다 (디코더 서브-네트워크). 일부 경우들에서, 디코더의 다른 서브-네트워크들이 존재할 수 있다. 이러한 딥 러닝 기반 시스템 아키텍처는 변환 플러스 양자화 모듈(transform plus quantization module)(또는 인코더 서브-네트워크)과 엔트로피 모델링 서브-네트워크 모듈의 조합으로 볼 수 있다.

[0032] 비디오 압축을 위한 대부분의 기존의 딥 러닝 기반 아키텍처는 RGB, YUV 4:4:4, 또는 다른 비-서브샘플링된 입력 포맷과 같은 비-서브샘플링된 입력 포맷에서 동작하도록 설계된다. 그러나, HEVC 및 VVC 와 같은 비디오 코딩 표준들은 그들 각각의 메인 프로파일들에서 YUV 4:2:0 컬러 포맷을 지원하도록 설계된다. 4:2:0 YUV 포맷을 지원하기 위해, 서브샘플링되지 않은 입력 포맷들에서 동작하도록 설계된 딥 러닝 기반 아키텍처들이 수정되어야 한다.

[0033] 하나 이상의 프레임들(예를 들어, 비디오 프레임) 중 하나의 컬러 컴포넌트를 사용하여 프레임의 컬러 컴포넌트 및 다른 컬러 컴포넌트에 대한 정보를 추정할 수 있는 ML-기반 시스템(예를 들어, 딥-러닝 기반 시스템)을 제공하는 시스템들, 장치들, 프로세스들(방법들이라고도 지칭됨), 및 컴퓨터-판독가능 매체들(집합적으로 "시스템들 및 기법들"이라고 지칭됨)이 본 명세서에 설명된다. 일부 양태들에서, ML-기반 시스템은 휘도-색차(YUV) 입력 포맷들을 갖는 입력 데이터를 프로세싱하도록 설계될 수 있다. 이러한 양태들에서, ML-기반 시스템은 루마 성분 및 하나 이상의 크로마 성분들 양자에 대한 모션 정보 (예를 들어, 플로우 정보, 예컨대 광학 플로우 정보) 를 추정하기 위해 (예를 들어, ML-기반 시스템에 의해 재구성된) 이전에-재구성된 프레임 및 현재 프레임 양자의 루마 성분을 사용할 수 있다. 일부 경우들에서, 루마 성분에 대한 모션 정보를 학습한 후, 다운 샘플링을 갖는 컨볼루션 계층은 하나 이상의 크로마 성분들에 대한 모션 정보 (예를 들어, 플로우 정보) 를 학습하는데 사용될 수 있다. 일부 경우들에서, 하나 이상의 크로마 성분들에 대한 모션 정보는 (예를 들어, 컨볼루션 계층을 사용하지 않고) 루마 성분에 대한 모션 정보를 직접 서브샘플링함으로써 획득될 수 있다. 이러한 기술은 프레임의 모든 구성요소에 대해 수행될 수 있다. 이러한 기법들을 사용하여, ML-기반 시스템은 잠재 데이터 또는 비트스트림의 일부로서 코딩된 크로마 정보를 가질 필요 없이 크로마 모션 정보 (예를 들어, 플로우 정보) 를 결정할 수 있다 (예를 들어, 크로마 정보와 함께 사이드 정보를 전송할 필요성을 감소시킨다).

[0034] 전술한 바와 같이, ML 기반 시스템은 YUV 입력 포맷을 갖는 입력 데이터를 처리하도록 설계될 수 있다. YUV 포맷은 휘도 채널(Y) 및 한 쌍의 색차 채널(U 및 V)을 포함한다. U 채널은 크로미넌스 (또는 크로마)-청색 채널로 지칭될 수 있고, U 채널은 크로미넌스 (또는 크로마)-적색 채널로 지칭될 수 있다. 일부 경우들에서, 루미넌스 (Y) 채널 또는 컴포넌트는 또한 루마 채널 또는 컴포넌트로 지칭될 수 있다. 일부 경우들에서, 색차(U 및 V) 채널들 또는 성분들은 또한 크로마 채널들 또는 성분들로 지칭될 수 있다. YUV 입력 포맷들은 특히 YUV 4:2:0, YUV 4:4:4, YUV 4:2:2을 포함할 수 있다. 일부 경우들에서, 본 명세서에 설명된 시스템들 및 기법들은 Y-크로마 블루 (Cb)-크로마 레드 (Cr) (YCbCr) 포맷, 레드-그린-블루 (RGB) 포맷, 및/또는 다른 포맷을 갖는 데이터와 같은 다른 입력 포맷들을 핸들링하도록 설계될 수 있다. 본원에 설명된 ML-기반 시스템은 다수의 프레임들을 포함하는 독립형 프레임들(이미지들이라고도 지칭됨) 및/또는 비디오 데이터를 인코딩 및/또는 디코딩할 수 있다.

[0035] 본 개시내용의 추가적인 세부사항들 및 추가적인 양상들이 도면들과 관련하여 설명될 것이다.

[0036] 도 1 은 본 명세서에 설명된 기능들 중 하나 이상을 수행하도록 구성된 중앙 처리 유닛(CPU)(102) 또는 멀티-코어 CPU를 포함할 수 있는 시스템-온-칩(SOC)(100)의 예시적인 구현을 도시한다. 다른 정보 중에서도, 파라미터

들 또는 변수들(예를 들어, 신경 신호들 및 시냅스 가중치들), 계산 디바이스와 연관된 시스템 파라미터들(예를 들어, 가중치들을 갖는 뉴럴 네트워크), 지연들, 주파수 빈 정보, 태스크 정보는 신경 프로세싱 유닛(NPU)(108)과 연관된 메모리 블록, CPU(102)와 연관된 메모리 블록, 그래픽 프로세싱 유닛(GPU)(104)과 연관된 메모리 블록, 디지털 신호 프로세서(DSP)(106)와 연관된 메모리 블록, 메모리 블록(118)에 저장될 수 있고/있거나, 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.

[0037] 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 맞게 조정된 부가 프로세싱 블록들을 포함할 수도 있다. 일 구현에서, NPU는 CPU(102), DSP(106) 및/또는 GPU(104)에서 구현된다. SOC (100) 는 또한 센서 프로세서 (114), 이미지 신호 프로세서들 (ISP들)(116), 및/또는 내비게이션 모듈 (120) 을 포함할 수 있으며, 이는 글로벌 포지셔닝 시스템을 포함할 수도 있다.

[0038] SOC (100) 는 ARM 명령 세트에 기초할 수도 있다. 본 개시의 일 양태에서, CPU(102)에 로딩된 명령어들은 입력 값과 필터 가중치의 곱셈 곱에 대응하는 룩업 테이블(LUT)에서 저장된 곱셈 결과를 검색하기 위한 코드를 포함할 수 있다. CPU(102)에 로딩된 명령어들은 또한 곱셈 곱의 룩업 테이블 히트가 검출될 때 곱셈 곱의 곱셈 연산 동안 곱셈기를 디스에이블하기 위한 코드를 포함할 수 있다. 또한, CPU(102)에 로딩된 명령어들은 곱셈 곱의 룩업 테이블 미스가 검출될 때 입력 값과 필터 가중치의 계산된 곱셈 곱을 저장하기 위한 코드를 포함할 수 있다.

[0039] 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 기법들을 사용하는 디바이스는 머신 학습 기반 기법들을 사용하여 비디오를 더 효율적으로 압축할 수 있고, 압축된 비디오를 다른 디바이스에 송신할 수 있고, 다른 디바이스는 본원에서 설명된 머신 학습 기반 기법들을 사용하여 압축된 비디오를 더 효율적으로 압축해제할 수 있다.

[0040] 전술한 바와 같이, 뉴럴 네트워크는 기계 학습 시스템의 예이고, 입력 계층, 하나 이상의 은닉 계층들, 및 출력 계층을 포함할 수 있다. 데이터는 입력 계층의 입력 노드들로부터 제공되고, 프로세싱은 하나 이상의 은닉 계층들의 은닉 노드들에 의해 수행되고, 출력은 출력 계층의 출력 노드들을 통해 생성된다. 딥 러닝 네트워크들은 통상적으로 다수의 은닉 계층들을 포함한다. 뉴럴 네트워크의 각각의 계층은 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 계층들 중 하나 이상의 노드들의 중요도를 표시하는 데 사용되는 하나 이상의 가중치들을 포함할 수 있다. 일부 경우들에서, 딥 러닝 네트워크는 일련의 많은 숨겨진 계층들을 가질 수 있으며, 초기 계층들은 입력의 단순하고 낮은 레벨 특성들을 결정하는 데 사용되고, 이후 계층들은 더 복잡하고 추상적인 특성들의 계층을 구축한다.

[0041] 딥 러닝 아키텍처는 특징들의 계위를 학습할 수도 있다. 예를 들어, 시각적 데이터로 제시되면, 제 1 계층은 입력 스트림에서, 에지들과 같은 비교적 간단한 특징들을 인식하는 것을 학습할 수도 있다. 다른 예에서, 청각적 데이터로 제시되면, 제 1 계층은 특정 주파수들에서의 스펙트럼 전력을 인식하는 것을 학습할 수도 있다. 제 1 계층의 출력을 입력으로서 취하는 제 2 계층은, 시각 데이터에 대한 간단한 형상들 또는 청각 데이터에 대한 사운드들의 조합들과 같은 특징들의 조합들을 인식하는 것을 학습할 수도 있다. 예를 들어, 상위 계층들은 시각적 데이터에서의 복잡한 형상들 또는 청각적 데이터에서의 단어들을 나타내는 것을 학습할 수도 있다. 여전히 상위 계층들은 공통 시각적 객체들 또는 구어체들을 인식하는 것을 학습할 수도 있다.

[0042] 딥 러닝 아키텍처들은 자연스러운 계위 구조를 갖는 문제들에 적용될 때 특히 잘 수행할 수도 있다. 예를 들어, 모터구동 차량들 (motorized vehicle) 의 분류는 휠들, 윈드쉴드들 및 다른 특징들을 인식하는 것을 먼저 학습하는 것으로 이익을 얻을 수 있다. 이러한 특징들은 자동차, 트럭, 및 비행기를 인식하기 위해 상이한 방식들로 상위 계층에서 조합될 수도 있다.

[0043] 뉴럴 네트워크들은 다양한 접속성 패턴들로 설계될 수도 있다. 피드-포워드 네트워크들에서, 정보는 하위 계층에서 상위 계층으로 전달되고, 주어진 계층에서의 각각의 뉴런은 상위 계층들에서의 뉴런들에 통신한다. 계위적 표현은 상술한 바와 같이, 피드-포워드 네트워크의 연속적인 계층들에 구축될 수도 있다. 뉴럴 네트워크들

은 또한 순환 (recurrent) 또는 피드백 (또한 하향식이라 함) 연결들을 가질 수도 있다. 순환 연결에서, 주어진 계층의 뉴런으로부터의 출력은 동일한 계층의 다른 뉴런으로 통신될 수도 있다. 순환 아키텍처는 시퀀스로 뉴럴 네트워크에 전달되는 입력 데이터 청크들 중 하나보다 많은 청크들에 걸쳐 있는 패턴들을 인식하는데 도움이 될 수도 있다. 주어진 계층의 뉴런에서 하위 계층의 뉴런으로의 연결은 피드백 (또는 하향식) 연결이라고 한다. 많은 피드백 연결들을 갖는 네트워크는 하이-레벨 개념의 인식이 입력의 특정 로우-레벨 특징들을 식별하는 것을 보조할 수도 있을 때 도움이 될 수도 있다.

[0044] 뉴럴 네트워크의 계층들 사이의 연결들은 완전히 연결되거나 로컬로 연결될 수도 있다. 도 2a 는 완전히 연결된 뉴럴 네트워크(202)의 예를 도시한다. 완전히 연결된 뉴럴 네트워크 (202) 에서, 제 1 계층에서의 뉴런은 제 2 계층에서의 모든 뉴런에 그의 출력을 통신할 수도 있으므로, 제 2 계층에서의 각각의 뉴런이 제 1 계층에서의 모든 뉴런으로부터 입력을 수신할 것이다. 도 2b 는 국부적으로 연결된 뉴럴 네트워크(204)의 예를 도시한다. 로컬로 연결된 뉴럴 네트워크 (204) 에서, 제 1 층에서의 뉴런은 제 2 계층에서의 제한된 수의 뉴런들에 연결될 수도 있다. 보다 일반적으로, 로컬로 연결된 뉴럴 네트워크 (204) 의 로컬로 연결된 계층은 계층에서의 각각의 뉴런이 동일하거나 유사한 접속성 패턴을 가질 것이지만, 상이한 값들 (예를 들어, 210, 212, 214, 및 216) 을 가질 수도 있는 연결 강도들을 갖도록 구성될 수도 있다. 로컬로 연결된 접속성 패턴은 상위 계층에서 공간적으로 별개의 수용 필드들을 발생할 수도 있는데, 이는 주어진 영역에서 상위 계층 뉴런들이 네트워크에 대한 총 입력의 제한된 부분의 특성들에 대한 훈련을 통해 튜닝되는 입력들을 수신할 수도 있기 때문이다.

[0045] 로컬로 연결된 뉴럴 네트워크의 일 예는 컨볼루션 뉴럴 네트워크이다. 도 2c 는 컨볼루션 뉴럴 네트워크(206)의 예를 도시한다. 컨볼루션 뉴럴 네트워크 (206) 는 제 2 계층에서의 각각의 뉴런에 대한 입력들과 연관된 연결 강도들이 공유되도록 (예를 들어, 208) 구성될 수도 있다. 컨볼루션 뉴럴 네트워크들은 입력들의 공간적 위치가 의미있는 문제들에 매우 적합할 수도 있다. 컨볼루션 뉴럴 네트워크(206)는 본 개시내용의 양태들에 따라, 비디오 압축 및/또는 압축해제의 하나 이상의 양태들을 수행하는 데 사용될 수 있다.

[0046] 컨볼루션 뉴럴 네트워크의 하나의 타입은 딥 컨볼루션 네트워크 (DCN) 이다. 도 2d 는 자동차-장착 카메라와 같은 이미지 캡처링 디바이스(230)로부터 입력된 이미지(226)로부터 시각적 피처들을 인식하도록 설계된 DCN(200)의 상세한 예를 예시한다. 본 예의 DCN (200) 은 교통 표지판 및 교통 표지판 상에 제공된 번호를 식별하도록 훈련될 수도 있다. 물론, DCN (200) 은 차선 마킹들을 식별하거나 신호등들을 식별하는 것과 같은 다른 태스크들을 위해 훈련될 수도 있다.

[0047] DCN (200) 은 지도 학습으로 훈련될 수도 있다. 훈련 동안, DCN (200) 은 속도 제한 표지판의 이미지 (226) 와 같은 이미지로 제시될 수도 있고, 그 후 순방향 패스가 출력 (222) 을 생성하기 위해 계산될 수도 있다. DCN (200) 은 특징 추출 섹션 및 분류 섹션을 포함할 수도 있다. 이미지 (226) 를 수신하면, 컨볼루션 계층 (232) 은 이미지 (226) 에 컨볼루션 커널들 (미도시) 을 적용하여 특징 맵들 (218) 의 제 1 세트를 생성할 수도 있다. 예로서, 컨볼루션 계층 (232) 에 대한 컨볼루션 커널은 28x28 특징 맵들을 생성하는 5x5 커널일 수도 있다. 본 예에서, 4개의 상이한 특징 맵이 특징 맵들의 제 1 세트 (218) 에서 생성되기 때문에, 4개의 상이한 컨볼루션 커널이 컨볼루션 계층 (232) 에서 이미지 (226) 에 적용되었다. 컨볼루션 커널들은 또한 필터들 또는 컨볼루션 필터들로 지칭될 수도 있다.

[0048] 특징 맵들의 제 1 세트 (218) 는 특징 맵들의 제 2 세트 (220) 를 생성하기 위해 최대 풀링 계층 (미도시) 에 의해 서브샘플링될 수도 있다. 최대 풀링 계층은 특징 맵들 (218) 의 제 1 세트의 사이즈를 감소시킨다. 즉, 14x14 와 같은 특징 맵들의 제 2 세트 (220) 의 사이즈는 28x28 과 같은 특징 맵들의 제 1 세트 (218) 의 사이즈보다 작다. 감소된 사이즈는 메모리 소비를 감소시키면서 후속 계층에 유사한 정보를 제공한다. 특징 맵들의 제 2 세트 (220) 는 추가로, 특징 맵들의 하나 이상의 후속 세트 (미도시) 를 생성하기 위해 하나 이상의 후속 컨볼루션 계층 (미도시) 을 통해 컨볼루션될 수도 있다.

[0049] 도 2d 의 예에서, 제 2 세트의 특징 맵(220)은 제 1 특징 벡터(224)를 생성하도록 컨볼루션된다. 또한, 제 1 특징 벡터 (224) 는 제 2 특징 벡터 (228) 를 생성하도록 추가로 컨볼루션된다. 제 2 특징 벡터 (228) 의 각각의 특징은 "표지판", "60" 및 "100" 과 같은 이미지 (226) 의 가능한 특징에 대응하는 수를 포함할 수도 있다. 소프트맥스 함수 (softmax function)(미도시) 는 제 2 특징 벡터 (228) 에서의 수들을 확률로 변환할 수도 있다. 이와 같이, DCN (200) 의 출력 (222) 은 하나 이상의 특징을 포함하는 이미지 (226) 의 확률이다.

[0050] 본 예에서, "부호" 및 "60"에 대한 출력(222)에서의 확률들은 "30", "40", "50", "70", "80", "90" 및 "100"과 같은 출력(222)의 다른 것들의 확률들보다 높다. 훈련 전에, DCN (200) 에 의해 생성된 출력 (222) 은 부정확할 가능성이 있다. 따라서, 출력 (222) 과 타겟 출력 사이에 에러가 계산될 수도 있다. 타겟 출력은 이미지

(226) 의 실측 자료(ground truth)(예를 들어, "표지판" 및 "60") 이다. DCN (200) 의 가중치들은 그 후 DCN (200) 의 출력 (222) 이 타겟 출력과 더 밀접하게 정렬되도록 조정될 수도 있다.

[0051] 가중치들를 조정하기 위해, 러닝 알고리즘은 가중치들에 대한 그래디언트 벡터를 계산할 수도 있다. 그래디언트는 가중치가 조정되었으면 에러가 증가 또는 감소할 양을 표시할 수도 있다. 최상위 계층에서, 그래디언트는 끝에서 두번째 계층에서의 활성화된 뉴런 및 출력 계층에서의 뉴런을 연결하는 가중치의 값에 직접 대응할 수도 있다. 하위 계층들에서, 그래디언트는 가중치들의 값 및 상위 계층들의 계산된 에러 그래디언트들에 의존할 수도 있다. 가중치들은 그 후 에러를 감소시키기 위해 조정될 수도 있다. 가중치를 조정하는 이러한 방식은 뉴럴 네트워크를 통한 "역방향 패스" 를 수반하기 때문에 "역 전파" 로 지칭될 수도 있다.

[0052] 실제로, 가중치들의 에러 그래디언트는 작은 수의 예들에 걸쳐 계산될 수도 있어서, 계산된 그래디언트는 실제 에러 그래디언트에 근사한다. 이러한 근사화 방법은 확률적 그래디언트 하강법 (stochastic gradient descent) 으로 지칭될 수도 있다. 확률적 그래디언트 하강법은 전체 시스템의 달성가능한 에러율이 감소하는 것을 멈출 때까지 또는 에러율이 타겟 레벨에 도달할 때까지 반복될 수도 있다. 학습 후에, DCN 은 새로운 이미지들을 제시받을 수도 있고, 네트워크를 통한 포워드 패스는 DCN 의 추론 또는 예측으로 고려될 수도 있는 출력 (222) 을 산출할 수도 있다.

[0053] DBN (deep belief network) 은 은닉된 노드들의 다중 계층들을 포함하는 확률 모델이다. DBN 은 훈련 데이터 세트의 계위적 표현을 추출하는데 사용될 수도 있다. DBN 은 제한된 볼츠만 머신 (Restricted Boltzmann Machines)(RBM) 의 계층들을 적층하여 획득될 수도 있다. RBM 은 입력들의 세트에 걸친 확률 분포를 학습할 수 있는 인공 뉴럴 네트워크의 타입이다. RBM들은 각각의 입력이 카테고리화되어야 하는 클래스에 관한 정보의 부재 시 확률 분포를 학습할 수 있기 때문에, RBM들은 종종 비지도 학습에 사용된다. 하이브리드 비지도 및 지도 패러다임을 사용하여, DBN 의 최하위 RBM들은 비지도 방식으로 훈련될 수도 있고 특징 추출기들로서 작용할 수도 있으며, 최상위 RBM 은 (이전 계층 및 타겟 클래스들로부터의 입력들의 공동 분포에 대해) 지도 방식으로 훈련될 수도 있고 분류기로서 작용할 수도 있다.

[0054] 딥 컨볼루션 네트워크 (DCN) 는 추가적인 풀링 및 정규화 계층들로 구성된, 컨볼루션 네트워크들의 네트워크들이다. DCN들은 많은 태스크들에 대해 최첨단 성능을 달성하였다. DCN들은 입력 및 출력 타겟들 양자 모두가 많은 예시들에 대해 알려져 있고 그래디언트 하강 방법들의 사용에 의해 네트워크의 가중치들을 수정하는데 사용되는 지도 학습을 사용하여 훈련될 수 있다.

[0055] DCN 은 피드-포워드 네트워크일 수도 있다. 또한, 상술한 바와 같이, DCN 의 제 1 계층에서의 뉴런으로부터 다음 상위 계층에서의 뉴런들의 그룹으로의 연결들은 제 1 계층에서의 뉴런들에 걸쳐 공유된다. DCN들의 피드-포워드 및 공유 연결들은 빠른 프로세싱을 위해 이용될 수도 있다. DCN 의 계산 부담은 예를 들어, 순환 또는 피드백 연결들을 포함하는 유사하게 사이징된 뉴럴 네트워크의 것보다 훨씬 적을 수도 있다.

[0056] 컨볼루션 네트워크의 각각의 계층의 프로세싱은 공간적으로 불변 템플릿 또는 기저 투영으로 간주될 수도 있다. 입력이 컬러 이미지의 적색, 녹색 및 청색 채널들과 같은 다중 채널들로 먼저 분해되면, 그 입력에 대해 훈련된 컨볼루션 네트워크는 이미지의 축들을 따라 2개의 공간 차원 및 컬러 정보를 캡처하는 제 3 차원을 갖는, 3 차원으로 간주될 수도 있다. 컨볼루션 연결들의 출력들은 후속 계층에서 특징 맵을 형성하는 것으로 간주될 수도 있고, 특징 맵의 각각의 엘리먼트 (예를 들어, 220) 는 이전 계층에서의 뉴런들의 범위 (예를 들어, 특징 맵들 (218)) 로부터 그리고 다중 채널들 각각으로부터 입력을 수신한다. 피처 맵에서의 값들은 교정 (rectification) 과 같은 비-선형성, max(0,x) 으로 추가로 프로세싱될 수도 있다. 인접한 뉴런들로부터의 값들은 추가로 풀링될 수도 있으며, 이는 다운 샘플링에 대응하고, 부가적인 로컬 불변 및 차원성 감소를 제공할 수도 있다.

[0057] 도 3 은 딥 컨볼루션 네트워크 (350) 의 일 예를 예시하는 블록도이다. 딥 컨볼루션 네트워크(350)는 연결성 및 가중치 공유에 기초하여 다수의 상이한 유형의 계층을 포함할 수 있다. 도 3 에 도시된 바와 같이, 딥 컨볼루션 네트워크(350)는 컨볼루션 블록들(354A, 354B)을 포함한다. 컨볼루션 블록들(354A, 354B) 각각은 컨볼루션 계층(CONV)(356), 정규화 계층(LNorm)(358), 및 최대 풀링 계층(MAX POOL)(360)으로 구성될 수 있다.

[0058] 컨볼루션 계층들(356)은 특징 맵을 생성하기 위해 입력 데이터(352)에 적용될 수 있는 하나 이상의 컨볼루션 필터를 포함할 수 있다. 블록들(354A, 354B) 상의 단지 2개의 컨볼루션(convolution)만이 도시되지만, 본 개시내용은 그렇게 제한되지 않으며, 대신에, 설계 선호도에 따라 임의의 수의 컨볼루션 블록들(예를 들어, 블록들(354A, 354B))이 딥 컨볼루션 네트워크(350)에 포함될 수 있다. 정규화 계층 (358) 은 컨볼루션 필터들의 출력

을 정규화할 수도 있다. 예를 들어, 정규화 계층 (358) 은 화이트닝 또는 측면 억제를 제공할 수도 있다. 최대 풀링 계층 (360) 은 로컬 불변 및 차원성 감소를 위해 공간에 걸쳐 다운 샘플링 집성을 제공할 수도 있다.

[0059] 예를 들어, 딥 컨볼루션 네트워크의 병렬 필터 뱅크들은 고성능 및 저전력 소비를 달성하기 위해 SOC (100) 의 CPU (102) 또는 GPU (104) 상에 로딩될 수도 있다. 대안적인 실시형태들에 있어서, 병렬 필터 뱅크들은 SOC (100) 의 DSP (106) 또는 ISP (116) 상에 로딩될 수도 있다. 또한, 딥 컨볼루션 네트워크 (350) 는 센서들 및 내비게이션에 각각 전용된, 센서 프로세서 (114) 및 내비게이션 모듈 (120) 과 같은 SOC (100) 상에 존재할 수도 있는 다른 프로세싱 블록들에 액세스할 수도 있다.

[0060] 딥 컨볼루션 네트워크(350)는 또한 계층(362A)("FC1"로 라벨링됨) 및 계층(362B)("FC2"로 라벨링됨)과 같은 하나 이상의 완전히 연결된 계층들을 포함할 수 있다. 딥 컨볼루션 네트워크 (350) 는 로지스틱 회귀 (LR) 계층 (364) 을 더 포함할 수도 있다. 딥 컨볼루션 네트워크(350)의 각각의 계층(356, 358, 360, 362A, 362B, 364) 사이에는 업데이트될 가중치들(도시되지 않음)이 있다. 계층들(예를 들어, 356, 358, 360, 362A, 362B, 364) 각각의 출력은 컨볼루션 블록들(354A) 중 제1 컨볼루션 블록에서 공급되는 입력 데이터(352)(예를 들어, 이미지들, 오디오, 비디오, 센서 데이터 및/또는 다른 입력 데이터)로부터 계층적 특징 표현들을 학습하기 위해 딥 컨볼루션 네트워크(350)에서 계층들(예를 들어, 356, 358, 360, 362A, 362B, 364) 중 후속하는 하나의 입력으로서 기능할 수 있다. 딥 컨볼루션 네트워크 (350) 의 출력은 입력 데이터 (352) 에 대한 분류 스코어 (366) 이다. 분류 스코어 (366) 는 확률들의 세트일 수도 있고, 여기서 각각의 확률은 특징들의 세트로부터의 특징을 포함하는, 입력 데이터의 확률이다.

[0061] 전술한 바와 같이, 디지털 비디오 데이터는 많은 양의 데이터를 포함할 수 있으며, 이는 통신 네트워크들뿐만 아니라 비디오 데이터를 처리하고 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 크기를 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 픽셀들의 해상도)는 프레임당 12.4 메가바이트, 또는 초당 297.6 메가바이트를 점유할 수 있다. 초당 24개의 프레임으로 4K 해상도로 기록된 채널당 압축되지 않은 16비트 비디오는 프레임당 49.8메가바이트, 즉 초당 1195.2메가바이트를 차지할 수 있다.

[0062] 네트워크 대역폭은 큰 비디오 파일이 문제가 될 수 있는 또 다른 제약이다. 예를 들어, 비디오 콘텐츠는 종종 무선 네트워크들(예를 들어, LTE, LTE-Advanced, New Radio(NR), WiFi TM, Bluetooth TM, 또는 다른 무선 네트워크들을 통해)을 통해 전달되고, 소비자 인터넷 트래픽의 큰 부분을 구성할 수 있다. 무선 네트워크들에서 이용가능한 대역폭의 양의 진보들에도 불구하고, 이들 네트워크들에서 비디오 콘텐츠를 전달하는데 사용되는 대역폭의 양을 감소시키는 것이 여전히 바람직할 수도 있다.

[0063] 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.

[0064] 비디오 콘텐츠의 크기 - 따라서 비디오 콘텐츠를 저장하는데 수반되는 저장의 양 - 및 비디오 콘텐츠를 전달하는데 수반되는 대역폭의 양을 감소시키기 위해, 다양한 비디오 코딩 기법들이 특히 HEVC, AVC, MPEG, VVC와 같은 특정 비디오 코딩 표준에 따라 수행될 수 있다. 비디오 코딩은 종종 인터-예측(inter-prediction) 또는 인트라-예측(intra-prediction)과 같은 예측 방법들을 사용하며, 이는 비디오 이미지들 또는 시퀀스들에 존재하는 리던던시들을 이용한다. 비디오 코딩 기법들의 공통 목적은 비디오 품질에 대한 열화들을 회피 또는 최소화하면서 더 낮은 비트 레이트를 사용하는 형태로 비디오 데이터를 압축하는 것이다. 비디오 서비스들에 대한 수요가 증가하고 새로운 비디오 서비스들이 이용가능해짐에 따라, 더 양호한 코딩 효율, 성능, 및 레이트 제어를 갖는 코딩 기법들이 필요하다.

[0065] 일반적으로 , 인코딩 디바이스는 인코딩된 비디오 비트스트림을 생성하기 위해 비디오 코딩 표준에 따라 비디오 데이터를 인코딩한다. 일부 예들에 있어서, 인코딩된 비디오 비트스트림 (또는 "비디오 비트스트림" 또는 "비트스트림") 은 일련의 하나 이상의 코딩된 비디오 시퀀스들이다. 인코딩 디바이스는 각각의 픽처를 다중의 슬라이스들로 파티셔닝함으로써 픽처들의 코딩된 표현들을 생성할 수 있다. 슬라이스는, 그 슬라이스에서의 정보가 동일한 픽처 내의 다른 슬라이스들로부터의 데이터에 의존하지 않고 코딩되도록 다른 슬라이스들에 독립적이다. 슬라이스는 독립적인 슬라이스 세그먼트, 및 만약 존재한다면, 이전 슬라이스 세그먼트들에 의존하는 하나 이상의 종속적인 슬라이스 세그먼트들을 포함하는 하나 이상의 슬라이스 세그먼트들을 포함한다. HEVC 에서,

슬라이스들은 루마 샘플들 및 크로마 샘플들의 코딩 트리 블록들 (CTB들) 로 파티셔닝된다. 루마 샘플들의 CTB 와 크로마 샘플들의 하나 이상의 CTB들이, 그 샘플들을 위한 신택스와 함께, 코딩 트리 유닛(coding tree unit, CTU)이라고 지칭된다. CTU 는 또한 "트리 블록" 또는 "최대 코딩 유닛" (largest coding unit; LCU) 으로 지칭될 수도 있다. CTU 는 HEVC 인코딩을 위한 기본 프로세싱 유닛이다. CTU 는 다양한 사이즈들의 다중 코딩 유닛들 (Cus) 로 분할될 수 있다. CU 는 코딩 블록들 (Cbs) 로 지칭되는 루마 및 크로마 샘플 어레이들을 포함한다.

[0066] 루마 및 크로마 CB 들은 예측 블록 (PB) 들로 더 분할될 수 있다. PB 는 (이용 가능하거나 사용을 위해 인에이블될 때) 인터 예측 또는 인트라 블록 카피 (IBC) 예측에 대해 동일한 모션 파라미터들을 사용하는 루마 성분 또는 크로마 성분의 샘플들의 블록이다. 루마 PB 및 하나 이상의 크로마 PB들은, 관련 구문과 함께, 예측 유닛 (PU) 을 형성한다. 인터-예측을 위해, 모션 파라미터들의 세트 (예컨대, 하나 이상의 모션 벡터들, 참조 인덱스들 등) 가 각각의 PU 에 대해 비트스트림으로 시그널링되고, 루마 PB 및 하나 이상의 크로마 PB들의 인터-예측을 위해 사용된다. 모션 파라미터들은 또한, 모션 정보로서 지칭될 수 있다. CB 는 또한 하나 이상의 변환 블록들 (Tbs) 로 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 신택스 엘리먼트들을 나타낸다. 변환 코딩은 하기에 보다 상세히 기재한다.

[0067] HEVC 표준에 따르면, 변환들은 TU들을 이용하여 수행될 수도 있다. TU들은 주어진 CU 내의 PU들의 크기에 기초하여 사이징될 수도 있다. TU들은 통상적으로 PU들과 동일한 크기이거나 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. TU들에 연관되는 픽셀 차이 값들이 변환 계수들을 형성하도록 변환될 수도 있다. 그 다음, 변환 계수들은 인코딩 디바이스에 의해 양자화될 수도 있다.

[0068] 일단 비디오 데이터의 픽처들이 CU들로 파티셔닝되면, 인코딩 디바이스는 예측 모드를 사용하여 각각의 PU 를 예측한다. 그 다음, 예측 유닛 또는 예측 블록은 잔차들을 얻기 위해 오리지널 비디오 데이터로부터 감산된다 (하기에서 설명됨). 각각의 CU 에 대해, 예측 모드는 신택스 데이터를 사용하여 비트스트림 내부에서 시그널링될 수도 있다. 예측 모드는 인트라 예측 (또는 인트라-화상 예측) 또는 인터 예측 (또는 인터-화상 예측) 을 포함할 수도 있다. 인트라 예측은 픽처 내에서 공간적으로 이웃하는 샘플 간의 상관 (correlation) 을 이용한다. 예를 들어, 인트라 예측을 사용하여, 각각의 PU 는, 예를 들어, PU 에 대한 평균값을 발견하기 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 픽처 내의 이웃하는 이미지 데이터로부터 예측된다. 인터 예측은 이미지 샘플들의 블록에 대한 모션 보상된 예측을 도출하기 위해 픽처들 간의 시간적 상관을 이용한다. 예를 들어, 인터 예측을 사용하여, 각각의 PU 는 (출력 순서로 현재 픽처의 전 또는 후의) 하나 이상의 레퍼런스 픽처들에서의 이미지 데이터로부터의 모션 보상 예측을 사용하여 예측된다. 인터 픽처 또는 인트라 픽처 예측을 사용하여 픽처 영역을 코딩할지 여부의 결정은 예를 들어 CU 레벨에서 행해질 수도 있다.

[0069] 인트라 및/또는 인터 예측을 이용하여 예측을 수행한 후, 인코딩 디바이스는 변환 및 양자화를 수행할 수 있다. 예를 들어, 예측 다음에, 인코딩 디바이스는 PU에 대응하는 잔차 값들을 계산할 수도 있다. 잔차 값들은 코딩되는 픽셀들의 현재 블록 (PU) 과 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 셋트를 포함한다. 일부 예들에서, 잔차 블록은 2 차원 블록 포맷 (예를 들어, 2 차원 매트릭스 또는 어레이의 픽셀 값들) 으로 표현될 수도 있다. 이러한 예에서, 잔차 블록은 픽셀 값들의 2 차원 표현이다.

[0070] 예측이 수행된 후에 남을 수 있는 임의의 잔차 데이터는 이산 코사인 변환, 이산 사인 변환, 정수 변환, 웨이브렛 변환, 다른 적절한 변환 함수 또는 이들의 임의의 조합에 기초할 수도 있는 블록 변환을 사용하여 변환된다. 일부 경우에서, 하나 이상의 블록 변환들 (예를 들어, 크기 32 x 32, 16 x 16, 8 x 8, 4 x 4, 또는 다른 적합한 사이즈) 이 각각의 CU 의 잔차 데이터에 적용될 수도 있다. 일부 실시형태들에서, TU 는 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.

[0071] 인코딩 디바이스는 변환 계수들의 양자화를 수행할 수 있다. 양자화는 변환 계수들을 양자화하여 그 계수들을 나타내는데 사용되는 데이터의 양을 감소시킴으로써 추가의 압축을 제공한다. 예를 들어, 양자화는 그 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 일례에 있어서, n 비트 값을 갖는 계수는 양자화 동안 m 비트 값으로 라운드-다운될 수도 있으며, 여기서, n 은 m 보다 크다.

[0072] 일단 양자화가 수행되면, 코딩된 비디오 비트스트림은 양자화된 변환 계수, 예측 정보 (예를 들어, 예측 모드, 모션 벡터, 블록 벡터 등), 파티셔닝 정보, 및 다른 구문 데이터와 같은 임의의 다른 적절한 데이터를 포함한다. 코딩된 비디오 비트스트림의 상이한 엘리먼트들은 그 후 인코딩 디바이스에 의해 엔트로피 인코딩될 수도 있다. 일부 예들에 있어서, 인코딩 디바이스는, 엔트로피 인코딩될 수 있는 직렬화된 벡터를 생성하기 위해 미리 정의된 스캔 순서를 활용하여 양자화된 변환 계수들을 스캐닝할 수도 있다. 일부 예들에서, 인코딩 디바이스는 적응적 스캔을 수행할 수도 있다. 벡터 (예를 들어, 1차원 벡터) 를 형성하기 위해 양자화된 변환 계수들을 스캐닝한 후, 인코딩 디바이스는 벡터를 엔트로피 인코딩할 수도 있다. 예를 들어, 인코딩 디바이스는 컨텍스트 적응적 가변 길이 코딩, 컨텍스트 적응적 이진 산술 코딩, 신택스 기반 컨텍스트 적응적 이진 산술 코딩, 확률 인터벌 파티셔닝 엔트로피 코딩, 또는 다른 적합한 엔트로피 인코딩 기법을 사용할 수도 있다.

[0073] 인코딩 디바이스는 인코딩된 비디오 비트스트림을 저장할 수 있고/있거나 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. 디코딩 디바이스는 그 후 인코딩된 비디오 비트스트림 데이터에 대해 리스케일링하고 역 변환을 수행할 수도 있다. 이어서, 잔차 데이터는 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 부가된다. 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.

[0074] 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 선험적으로 정의될 수도 있다. 그러나, 일부 경우들에서, 기계 학습(ML) 기반 이미지 및/또는 비디오 시스템은 딥 러닝 기반 엔드-투-엔드 비디오 코딩(DLEC) 시스템과 같은 비-ML 기반 이미지 및 비디오 코딩 시스템들에 비해 이점들을 제공할 수 있다. 전술한 바와 같이, 많은 딥 러닝 기반 시스템들은 오토인코더 서브-네트워크(인코더 서브-네트워크) 및 엔트로피 코딩에 사용되는 양자화된 레이턴시들에 대해 확률 모델을 학습하는 것을 담당하는 제2 서브-네트워크의 조합으로서 설계된다. 이러한 아키텍처는 변환, 양자화 모듈(인코더 서브-네트워크) 및 엔트로피 모델링 서브-네트워크 모듈의 조합으로 간주될 수 있다.

[0075] 도 4는 딥 러닝 기반 시스템(410)을 사용하여 비디오 인코딩 및 디코딩을 수행하도록 구성된 디바이스(402)를 포함하는 시스템(400)을 도시한다. 디바이스(402)는 카메라(407) 및 저장 매체(414)(예를 들어, 데이터 저장 디바이스)에 결합된다. 일부 구현예에서, 카메라(407)는 딥 러닝 기반 시스템(410)에 의한 인코딩을 위해 이미지 데이터(408)(예를 들어, 비디오 데이터 스트림)를 프로세서(404)에 제공하도록 구성된다. 일부 구현들에서, 디바이스(402)는 다수의 카메라들(예를 들어, 듀얼 카메라 시스템, 3개의 카메라들, 또는 다른 수의 카메라들)에 결합될 수 있고/있거나 이들을 포함할 수 있다. 일부 경우들에서, 디바이스(402)는 마이크로폰 및/또는 다른 입력 디바이스(예를 들어, 키보드, 마우스, 터치스크린 및/또는 터치패드와 같은 터치 입력 디바이스, 및/또는 다른 입력 디바이스)에 결합될 수 있다. 일부 예들에서, 카메라(407), 저장 매체(414), 마이크로폰, 및/또는 다른 입력 디바이스는 디바이스(402)의 일부일 수 있다.

[0076] 디바이스 (402) 는 또한 하나 이상의 무선 네트워크들, 하나 이상의 유선 네트워크들, 또는 이들의 조합과 같은 송신 매체 (418) 를 통해 제 2 디바이스 (490)에 커플링된다. 예를 들어, 전송 매체(418)는 무선 네트워크, 유선 네트워크, 또는 유선 및 무선 네트워크의 조합에 의해 제공되는 채널을 포함할 수 있다. 전송 매체(418)는 패킷 기반 네트워크, 예컨대 로컬 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크의 부분을 형성할 수도 있다. 전송 매체(418)는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스로부터 수신 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, WiFiTM, RF (radio frequency), UWB, WiFi-Direct, 셀룰러, LTE(Long-Term Evolution), WiMaxTM 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 수 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라 변조되고, 수신 디바이스로 송신될 수도 있다.

[0077] 디바이스(402)는 메모리(406), 제1 인터페이스("I/F 1")(412) 및 제2 인터페이스("I/F 2")(416)에 결합된 하나 이상의 프로세서(본 명세서에서 "프로세서"로 지칭됨)(404)를 포함한다. 프로세서(404)는 카메라(407)로부터, 메모리(406)로부터, 및/또는 저장 매체(414)로부터 이미지 데이터(408)를 수신하도록 구성된다. 프로세서(404)는 제 1 인터페이스(412)를 통해(예를 들어, 메모리 버스를 통해) 저장 매체(414)에 커플링되고, 제 2 인터페이스(416)(예를 들어, 네트워크 인터페이스 디바이스, 무선 트랜시버 및 안테나, 하나 이상의 다른 네트워크 인터페이스 디바이스들, 또는 이들의 조합)를 통해 송신 매체(418)에 커플링된다.

[0078] 프로세서(404)는 딥 러닝 기반 시스템(410)을 포함한다. 딥 러닝 기반 시스템(410)은 인코더 부분(462) 및 디코더 부분(466)을 포함한다. 일부 구현예에서, 딥 러닝 기반 시스템(410)은 하나 이상의 오토인코더를 포함할 수 있다. 인코더 부분(462)은 입력 데이터(470)를 수신하고 입력 데이터(470)를 처리하여 입력 데이터(470)에 적어도 부분적으로 기초하여 출력 데이터(474)를 생성하도록 구성된다.

[0079] 일부 구현예에서, 딥 러닝 기반 시스템(410)의 인코더 부분(462)은 출력 데이터(474)를 생성하기 위해 입력 데이터(470)의 손실 압축을 수행하도록 구성되어, 출력 데이터(474)는 입력 데이터(470)보다 적은 비트를 갖는다. 인코더 부분(462)은 임의의 이전 표현들(예를 들어, 하나 이상의 이전에 재구성된 프레임들)에 기초하여 모션 보상을 사용하지 않고 입력 데이터(470)(예를 들어, 이미지들 또는 비디오 프레임들)를 압축하도록 트레이닝될 수 있다. 예를 들어, 인코더 부분(462)은 이전에 재구성된 프레임들의 임의의 데이터를 사용하지 않고 그 비디오 프레임으로부터만 비디오 데이터를 사용하여 비디오 프레임을 압축할 수 있다. 인코더 부분 (462)에 의해 프로세싱된 비디오 프레임들은 본 명세서에서 인트라-예측된 프레임 (I-프레임들) 으로서 지칭될 수 있다. 일부 예들에서, I-프레임들은 (예를 들어, HEVC, VVC, MPEG-4, 또는 다른 비디오 코딩 표준에 따라) 전통적인 비디오 코딩 기법들을 사용하여 생성될 수 있다. 이러한 예들에서, 프로세서 (404) 는 HEVC 표준에 대하여 위에서 설명된 것과 같은 블록-기반 인트라 예측을 수행하도록 구성된 비디오 코딩 디바이스 (예를 들어, 인코딩 디바이스) 를 포함하거나 또는 이와 커플링될 수도 있다. 이러한 예들에서, 딥 러닝 기반 시스템(410)은 프로세서(404)로부터 배제될 수 있다.

[0080] 일부 구현에서, 딥 러닝 기반 시스템(410)의 인코더 부분(462)은 이전 표현(예를 들어, 하나 이상의 이전에 재구성된 프레임)에 기초한 모션 보상을 사용하여 입력 데이터(470)(예를 들어, 비디오 프레임)를 압축하도록 트레이닝될 수 있다. 예를 들어, 인코더 부분(462)은 그 비디오 프레임으로부터의 비디오 데이터를 사용하여 그리고 이전에 재구성된 프레임들의 데이터를 사용하여 비디오 프레임을 압축할 수 있다. 인코더 부분 (462)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 위치들로 어떻게 이동하는지를 설명함으로써 현재 프레임의 데이터를 결정하는데 사용될 수 있다.

[0081] 도시된 바와 같이, 딥 러닝 기반 시스템(410)의 인코더 부분(462)은 뉴럴 네트워크(463) 및 양자화기(464)를 포함할 수 있다. 뉴럴 네트워크(463)는 하나 이상의 컨볼루션 뉴럴 네트워크(CNN), 하나 이상의 완전-연결 뉴럴 네트워크, 하나 이상의 게이트 순환 유닛(GRU), 하나 이상의 장기 단기 메모리(LSTM) 네트워크, 하나 이상의 ConvRNN, 하나 이상의 ConvGRU, 하나 이상의 ConvLSTM, 하나 이상의 GAN, 이들의 임의의 조합, 및/또는 중간 데이터(472)를 생성하는 다른 유형의 뉴럴 네트워크 아키텍처를 포함할 수 있다. 중간 데이터(472)는 양자화기(464)에 입력된다. 인코더 부분 (462) 에 포함될 수도 있는 컴포넌트들의 예들이 도 6 - 도 10 에 예시되어 있다.

[0082] 양자화기(464)는 출력 데이터(474)를 생성하기 위해 중간 데이터(472)의 양자화 및 일부 경우들에서 엔트로피 코딩을 수행하도록 구성된다. 출력 데이터(474)는 양자화된(및 일부 경우에 엔트로피 코딩된) 데이터를 포함할 수 있다. 양자화기(464)에 의해 수행되는 양자화 동작들은 중간 데이터(472)로부터 양자화된 코드들(또는 딥 러닝 기반 시스템(410)에 의해 생성된 양자화된 코드들을 나타내는 데이터)의 생성을 야기할 수 있다. 양자화 코드들(또는 양자화된 코드들을 나타내는 데이터)은 또한 잠재 코드들(latent codes) 또는 잠재(latent)(z로 표

시됨)로 지칭될 수 있다. 잠재성에 적용되는 엔트로피 모델은 본 명세서에서 "프라이어"로 지칭될 수 있다. 일부 예들에서, 양자화 및/또는 엔트로피 코딩 동작들은 기존의 비디오 코딩 표준들에 따라 비디오 데이터를 인코딩 및/또는 디코딩할 때 수행되는 기존의 양자화 및 엔트로피 코딩 동작들을 사용하여 수행될 수 있다. 일부 예들에서, 양자화 및/또는 엔트로피 코딩 동작들은 딥 러닝 기반 시스템(410)에 의해 수행될 수 있다. 하나의 예시적인 예에서, 딥 러닝 기반 시스템(410)은 지도 트레이닝(supervised training)을 사용하여 트레이닝될 수 있으며, 잔차 데이터는 입력으로서 사용되고, 양자화된 코드들 및 엔트로피 코드들은 트레이닝 동안 알려진 출력(라벨들)으로서 사용된다.

[0083] 딥 러닝 기반 시스템(410)의 디코더 부분(466)은 (예를 들어, 양자화기(464)로부터 및/또는 저장 매체(414)로부터 직접) 출력 데이터(474)를 수신하도록 구성된다. 디코더 부분(466)은 출력 데이터(474)를 처리하여 출력 데이터(474)에 적어도 부분적으로 기초하여 입력 데이터(470)의 표현(476)을 생성할 수 있다. 일부 예에서, 딥 러닝 기반 시스템(410)의 디코더 부분(466)은 하나 이상의 CNN, 하나 이상의 완전 연결 뉴럴 네트워크, 하나 이상의 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 다른 유형의 뉴럴 네트워크 아키텍처를 포함할 수 있는 신경망(468)을 포함한다. 디코더 부분(466)에 포함될 수 있는 컴포넌트들의 예들이 도 6 - 도 10 에 예시된다.

[0084] 프로세서 (404) 는 출력 데이터 (474) 를 송신 매체 (418) 또는 저장 매체 (414) 중 적어도 하나에 전송하도록 구성된다. 예를 들어, 출력 데이터(474)는 재구성된 데이터로서 입력 데이터(470)의 표현(476)을 생성하기 위해 디코더 부분(466)에 의한 나중의 검색 및 디코딩(또는 압축해제)을 위해 저장 매체(414)에 저장될 수 있다. 재구성된 데이터는 출력 데이터(474)를 생성하기 위해 인코딩/압축된 비디오 데이터의 재생을 위한 것과 같은 다양한 목적을 위해 사용될 수 있다. 일부 구현들에서, 출력 데이터 (474) 는 재구성된 데이터로서 입력 데이터 (470) 의 표현 (476) 을 생성하기 위해 디코더 부분 (466)에 매칭하는 다른 디코더 디바이스에서 (예를 들어, 디바이스 (402)에서, 제 2 디바이스 (490)에서, 또는 다른 디바이스에서) 디코딩될 수도 있다. 예를 들어, 제 2 디바이스 (490) 는 디코더 부분 (466) 과 매칭 (또는 실질적으로 매칭) 하는 디코더를 포함할 수도 있고, 출력 데이터 (474) 는 송신 매체 (418) 를 통해 제 2 디바이스 (490) 로 송신될 수도 있다. 제 2 디바이스(490)는 입력 데이터(470)의 표현(476)을 재구성된 데이터로서 생성하기 위해 출력 데이터(474)를 처리할 수 있다.

[0085] 시스템 (400) 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 있고 및/또는 이들을 이용하여 구현될 수 있고, 및/또는 본원에 기술된 다양한 동작들을 수행하기 위해 컴퓨터 소프트웨어, 펌웨어, 또는 이들의 임의의 조합을 포함할 수 있고 및/또는 이들을 이용하여 구현될 수 있다.

[0086] 시스템 (400) 이 특정 컴포넌트들을 포함하는 것으로 도시되지만, 당업자는 시스템 (400) 이 도 4 에 도시된 컴포넌트들보다 더 많거나 더 적은 컴포넌트들을 포함할 수 있다는 것을 이해할 것이다. 예를 들어, 시스템(400)은 또한 입력 디바이스 및 출력 디바이스(도시되지 않음)를 포함하거나 이를 포함하는 컴퓨팅 디바이스의 일부일 수 있다. 일부 구현들에서, 시스템(400)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 통신하고 그리고/또는 그에 전기적으로 접속되는 하나 이상의 프로세싱 디바이스들(예를 들어, 하나 이상의 CPU들, GPU들, 및/또는 다른 프로세싱 디바이스들), 무선 통신들을 수행하기 위한 하나 이상의 무선 인터페이스들(예를 들어, 각각의 무선 인터페이스에 대한 기저대역 프로세서 및 하나 이상의 트랜시버들을 포함함), 하나 이상의 하드와이어드 연결들을 통해 통신들을 수행하기 위한 하나 이상의 유선 인터페이스들(예를 들어, 직렬 인터페이스, 이를테면 범용 직렬 버스(USB) 입력, 라이트닝 커넥터, 및/또는 다른 유선 인터페이스), 및/또는 도 4 에 도시되지 않은 다른 컴포넌트들을 포함하는 컴퓨팅 디바이스를 포함할 수 있거나, 또는 그 일부일 수 있다.

[0087] 일부 구현들에서, 시스템(400)은 컴퓨팅 디바이스에 의해 로컬로 구현되고 그리고/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 기술들을 수행하기 위한

자원 능력들을 갖는 임의의 다른 컴퓨팅 디바이스를 포함할 수 있다.

[0088] 일 예에서, 딥 러닝 기반 시스템(410)은 프로세서(404)에 결합되고 프로세서(404)에 의해 실행 가능한 명령어들을 저장하도록 구성된 메모리(406), 및 안테나 및 프로세서(404)에 결합되고 출력 데이터(474)를 원격 디바이스에 송신하도록 동작 가능한 무선 트랜시버를 포함하는 휴대용 전자 디바이스에 통합될 수 있다.

[0089] 위에서 언급된 바와 같이, 딥 러닝 기반 시스템들은 통상적으로 RGB 또는 YUV 4:4:4과 같은 비-서브샘플링된 입력 포맷들을 프로세싱하도록 설계된다. RGB 입력을 타겟으로 하는 이미지 및 비디오 코딩 방식들의 예들은 J. Balle, D. Minnen, S. Singh, S. J. Hwang, N. Johnston, "Variational image compression with a scale hyperprior", ICLR, 2018("J. Balle Paper"로 지칭됨) 및 D. Minnen, J. Balle, G. Toderici, "Joint Autoregressive and Hierarchical Priors for Learned Image Compression", CVPR 2018("D. Minnen Paper"로 지칭됨)에 기술되어 있으며, 이는 그 전체로서 그리고 모든 목적을 위해 본원에 참고로 포함된다.

[0090] 도 5는 딥러닝 기반 시스템(500)의 일 예를 나타낸 도면이다. 도 5의 딥 러닝 기반 시스템에서의 g_a 및 g_s 서브 네트워크들은 각각 인코더 서브 네트워크(예를 들어, 인코더 부분(462)) 및 디코더 서브 네트워크(예를 들어, 디코더 부분(466))에 대응한다. 도 5 의 g_a 및 g_s 서브-네트워크들은 3-채널 RGB 입력을 위해 설계되고, 여기서 3개의 R, G, 및 B 입력 채널들 모두가 통과하고 동일한 뉴럴 네트워크 계층들(컨볼루션 계층들 및 일반화된 분할 정규화(GDN) 계층들)에 의해 프로세싱된다. 뉴럴 네트워크 계층들은 컨볼루션 연산들을 수행하는 컨볼루션 계층들(컨볼루션 계층(510)을 포함함) 및 로컬 분할 정규화를 구현하는 GDN 및/또는 IGDN(inverse-GDN) 비선형 계층들을 포함할 수 있다. 로컬 분할 정규화는 이미지의 밀도 모델링 및 압축에 특히 적합한 것으로 나타난 변환 유형이다. 딥 러닝 기반 시스템(도 5에 도시된 것과 같음)은 RGB 데이터(상이한 R, G 및 B 채널의 통계적 속성이 유사함)와 같은 유사한 통계적 특성을 갖는 입력 채널을 대상으로 한다.

[0091] 많은 딥 러닝-기반 시스템들이 RGB 입력을 프로세싱하도록 설계되지만, 대부분의 이미지 및 비디오 코딩 시스템들은 YUV 입력 포맷들(예를 들어, 많은 경우들에서 YUV 4:2:0 입력 포맷)을 사용한다. YUV 포맷의 데이터의 색차(U 및 V) 채널들은 휘도(Y) 채널에 대해 서브샘플링될 수 있다. 서브샘플링은 시각적 품질에 대한 최소한의 영향을 초래한다(예를 들어, 시각적 품질에 대한 유의하거나 눈에 띄는 영향이 없다). 서브샘플링된 포맷들은 YUV 4:2:0 포맷, YUV 4:2:2 포맷, 및/또는 다른 YUV 포맷들을 포함한다. 채널들에 걸친 상관은 YUV 포맷에서 감소되며, 이는 다른 컬러 포맷들(예를 들어, RGB 포맷)의 경우가 아닐 수 있다. 또한, 휘도(Y) 및 색차(U 및 V) 채널들의 통계치들은 상이하다. 예를 들어, U 및 V 채널들은 휘도 채널에 비해 더 작은 분산을 갖는 반면, 예를 들어, RGB 포맷들에서, 상이한 R, G, 및 B 채널들의 통계적 속성들은 더 유사하다. 비디오 코더들-디코더들 (또는 코덱들) 은 데이터의 입력 특성들에 따라 설계된다 (예를 들어, 코덱은 데이터의 입력 포맷에 따라 데이터를 인코딩 및/또는 디코딩할 수 있다). 예를 들어, 프레임의 색차 채널들이 서브샘플링되면(예를 들어, 색차 채널들이 휘도 채널과 비교하여 해상도의 절반임), CODEC이 모션 보상을 위해 프레임의 블록을 예측할 때, 휘도 블록은 색차 블록들과 비교하여 폭 및 높이 둘 모두에 대해 2배만큼 클 것이다. 다른 예에서, CODEC은, 다른 것들 중에서도, 얼마나 많은 픽셀들이 크로미넌스 및 휘도에 대해 인코딩되거나 디코딩될 것인지를 결정할 수 있다.

[0092] YUV 포맷들(예를 들어, YUV 4:2:0 포맷)을 지원하기 위해, 딥 러닝 기반 아키텍처들이 수정되어야 한다. 예를 들어, RGB 입력 데이터(위에서 언급한 바와 같이, 대부분의 딥 러닝 기반 시스템이 처리하도록 설계됨)가 YUV 4:4:4 입력 데이터(모든 채널이 동일한 차원을 가짐)로 대체되면, 입력 데이터를 처리하는 딥 러닝 기반 시스템의 성능은 휘도(Y) 및 색차(U 및 V) 채널의 상이한 통계적 특성으로 인해 감소된다. 전술한 바와 같이, 색차(U 및 V) 채널들은 YUV 4:2:0의 경우와 같은 일부 YUV 포맷들에서 서브샘플링된다. 예를 들어, YUV 4:2:0 포맷을 갖는 콘텐츠에 대해, U 및 V 채널 해상도는 Y 채널 해상도의 절반이다(U 및 V 채널들은 폭 및 높이가 절반으로 되기 때문에 Y 채널의 4분의 1인 크기를 갖는다). 이러한 서브샘플링은 입력 데이터가 딥 러닝 기반 시스템의 입력과 호환되지 않게 할 수 있다. 입력 데이터는 딥 러닝 기반 시스템이 인코딩 및/또는 디코딩하려고 시도하고 있는 정보(예를 들어, 휘도(Y) 및 색차(U 및 V) 채널들을 포함하는 3개의 채널들을 포함하는 YUV 프레임)이다.

[0093] 일부 엔드-투-엔드 비디오 코딩 딥 러닝 기반 시스템들에서, 오토인코더들은 오리지널 프레임들에 대한 인트라 프레임, 모션 벡터들(예를 들어, 조밀한 광학 흐름), 및 모션 보상된 프레임들의 잔차를 코딩하기 위해 사용된다. 일 예에서, 플로우 오토인코더는 광학 플로우 뿐만 아니라 스케일-공간을 코딩하기 위해 공동으로 학습하는데 사용될 수 있고, 레지듀얼 오토인코더는 와핑된 예측 프레임과 원본 프레임 사이의 레지듀얼을 모두 RGB

도메인에서 코딩한다.

[0094] 전술한 바와 같이, 하나 이상의 YUV 포맷(예를 들어, YUV 4:2:0 포맷)을 효율적으로 지원하는 ML 기반 시스템(예를 들어, 하나 이상의 딥 러닝 기반 아키텍처를 포함함)을 제공하는 시스템 및 기술이 본 명세서에 설명된다. 딥 러닝 기반 아키텍처(들)는 독립형 프레임들(또는 이미지들) 및/또는 다수의 프레임들을 포함하는 비디오 데이터를 인코딩 및/또는 디코딩할 수 있다. 예를 들어, ML-기반 시스템은 입력으로서, ML-기반 시스템의 이전 인스턴스에 의해 재구성될 수 있는, 현재 프레임의 루마 성분 및 이전에-재구성된 프레임의 루마 성분을 획득할 수 있다. ML-기반 시스템은 현재 프레임의 루마 성분에 대한 모션 정보(예를 들어, 광학 흐름 정보와 같은 흐름 정보)를 추정하기 위해 현재 및 이전 프레임들의 루마 성분들을 프로세싱할 수 있다. 현재 프레임의 루마 성분을 사용하여, ML-기반 시스템은 현재 프레임의 하나 이상의 크로마 성분들에 대한 모션 추정(예를 들어, 광학 흐름 정보와 같은 흐름 정보)을 결정(예를 들어, 추정)할 수 있다. 이러한 기술은 프레임의 모든 구성요소에 대해 수행될 수 있다. 더 자세한 내용은 후술한다.

[0095] 도 6은 비디오 코딩을 수행하도록 구성된 딥 러닝 기반 시스템(600)의 뉴럴 네트워크 아키텍처의 예를 예시하는 도면이다. 도 6의 뉴럴 네트워크 아키텍처는 인트라 예측 엔진(602) 및 인터 예측 엔진(610)을 포함한다. 인트라 예측 엔진(602) 및 인터 예측 엔진(610)은 도 6에 도시된 바와 같이 오토인코더들(예를 들어, 가변 오토인코더들(VAE))을 포함할 수 있지만, 다른 구현들에서 다른 타입들의 뉴럴 네트워크 아키텍처들을 포함할 수 있다. 도시된 바와 같이, 인트라 예측 엔진(602)은 입력 프레임(604)의 잠재 표현($\hat{\boldsymbol{y}}$로서 도시됨)을 생성하기 위해 입력 프레임(604)의 픽셀 정보를 프로세싱한다. 입력 프레임(604)은 입력 프레임(604)의 각각의 픽셀에 대해 루마 컴포넌트($X_Y^{(0)}$ 로서 도시됨) 및 2개의 크로마 컴포넌트들($X_U^{(0)}$ 및 $X_V^{(0)}$로서 도시됨)을 포함한다. 잠재 표현은 또한 입력 프레임 (604) 의 코딩된 버전인 다수의 비트들을 포함하는 비트스트림으로 지칭될 수 있다. 잠재 표현 $\hat{\boldsymbol{y}}$(또는 다른 디바이스로부터 수신된 잠재 표현/비트스트림)에 기초하여, 인트라-예측 엔진(602) 의 디코더 서브-네트워크는 입력 프레임 (604) 의 재구성된 버전인 재구성된 프레임 (606) ($\hat{X}_Y^{(0)}$, $\hat{X}_U^{(0)}$, $\hat{X}_V^{(0)}$로서 도시됨, 여기서 컴포넌트들에 대한 "햇(hat)"는 재구성된 값들을 표시함) 을 생성할 수 있다.

[0096] 인터-예측 엔진(610)은 플로우 엔진(618), 레지듀얼 엔진(620), 및 워핑 엔진(622)을 포함한다. 도시된 바와 같이, 플로우 엔진(618)은 입력으로서 (시간 t에서) 현재 프레임(614)의 루마 성분($X_Y^{(t)}$)과 (이전 시간 t-1에서) 이전 프레임(615)의 재구성된 루마 성분($\hat{X}_Y^{(t-1)}$)을 획득한다. 루마 성분($X_Y^{(t)}$) 및 루마 성분($\hat{X}_Y^{(t-1)}$)을 사용하여, 플로우 엔진(618)은 현재 프레임(614)에 대한 루마 성분($X_Y^{(t)}$)에 대한 모션 정보(예를 들어, 플로우 정보)의 잠재적 표현($\hat{\boldsymbol{y}}_f$)을 생성한다. 모션 정보는 (시간 t-1에서) 이전 프레임 (615)에 대한 (시간 t에서의) 현재 프레임 (614) 의 픽셀들의 이동을 표시하는 광학적 흐름 정보 (예를 들어, 복수의 모션 또는 변위 벡터들 및 일부 경우들에서는 픽셀 또는 샘플 당 스케일 성분) 를 포함할 수도 있다. 잠재적 표현($\hat{\boldsymbol{y}}_f$)은 또한 비트스트림으로 지칭될 수 있고, 현재 프레임 (614)에 대한 루마 성분($X_Y^{(t)}$)의 코딩된 버전을 표현하는 다수의 비트들을 포함할 수 있다. 플로우 엔진 (618) 이 크로마 성분들이 아니라 현재 프레임 (614)에 대한 루마 성분($X_Y^{(t)}$)을 프로세싱하기 때문에, 잠재적 표현($\hat{\boldsymbol{y}}_f$)(비트스트림) 은 모션 정보를 결정하기 위해 현재 프레임 (614) 의 모든 컴포넌트들을 사용하는 것과 비교하여 사이즈가 감소된다.

[0097] 루마 성분($X_Y^{(t)}$)의 잠재적 표현($\hat{\boldsymbol{y}}_f$)(또는 프레임의 컴포넌트를 표현하는 다른 디바이스로부터 수신된 비트스트림)을 사용하여, 플로우 엔진(618)은 현재 프레임(614)의 루마 성분($X_Y^{(t)}$)에 대한 모션 정보(f^L 로 표시됨)를 결정하고, 또한 현재 프레임(614)의 크로마 성분들($X_U^{(t)}$, $X_V^{(t)}$)에 대한 모션 정보(f^C 로 표시됨)를 결정한다. 루마 성분에 대한 결정된 모션 정보에 기초하여 크로마 성분에 대한 모션 정보를 결정 또는 추정하는 세부사항들은 이하에서 도 7a 및 7b 를 참조하여 설명된다.

[0098] 워핑 엔진 (622) 은 (시간 t에서) 현재 프레임 (614) 의 루마 성분($X_Y^{(t)}$) 및 크로마 성분($X_U^{(t)}$, $X_V^{(t)}$)에 대해 결정된 모션 정보 (f^L 및f^C)를 사용하여 워핑을 수행하도록 구성된다. 예를 들어, 워핑 엔진 (622) 은 현재 프레임 (614)에 대한 루마 성분($X_Y^{(t)}$) 및 크로마 성분($X_U^{(t)}$, $X_V^{(t)}$)의 모션 정보 (f^L 및 f^C)에 의해 표시된 양만큼 (시간 t에서) 현재 프레임 (614) 의 픽셀들을 워핑할 수 있다. 일부 양태들에서, 워핑 엔진(622)은 공간-스케일 흐름(SSF) 워핑을 수행할 수 있다. 예를 들어, SSF 워핑은 학습된 스케일-흐름 벡터들로부터 프레임간 예측들을 생성하기 위해 삼선형 보간을 적용할 수 있으며, 여기서 예측자들은 다음과 같이 공식화될 수 있다:

[0099] $$\boldsymbol{p_Y} \coloneqq trilinear(\boldsymbol{x_Y}, \boldsymbol{f^L})$$

[0100] $$\boldsymbol{p_U} \coloneqq trilinear(\boldsymbol{x_U}, \boldsymbol{f^C})$$, 및 $$\boldsymbol{p_V} \coloneqq trilinear(\boldsymbol{x_V}, \boldsymbol{f^C})$$

[0101] 따라서:

[0102] $$p_Y[x,y] = x_Y\left[x + v_x^L[x,y], y + v_y^L[x,y], s^L[x,y]\right]$$

[0103] $$p_U[x,y] = x_U\left[x + v_x^C[x,y], y + v_y^C[x,y], s^C[x,y]\right]$$

[0104] $$p_V[x,y] = x_V\left[x + v_x^C[x,y], y + v_y^C[x,y], s^C[x,y]\right]$$

[0105] 식 (1)

[0106] 위의 삼선형 보간은 루마 성분($X_Y^{(t)}$) 및 크로마 성분($X_U^{(t)}$)의 모션 정보 (f^L 및 f^C)에 기초하여 결정된 하나 이상의 워핑 파라미터들에 기초하여 (예를 들어, 각각의 루마 성분에 대해 그리고 각각의 개별 U 및 V 크로마 성분에 대해) 성분 단위로 수행될 수 있다. 예를 들어, 워핑 파라미터들은 모션 또는 변위 벡터의 (x-방향으로의) 수평 성분을 표현하는 v_x^*, 모션 또는 변위 벡터의 (y-방향으로의) 수직 성분을 표현하는 v_y^*, 공간 모션/변위 정보 (vx 및 vy) 와 결합되는 재구성된 프레임들의 점진적으로 평활화된 버전을 표현하는 (스케일 필드로 지칭되는) $\boldsymbol{s}$ 를 포함할 수도 있다.

[0107] (워핑 엔진 (622)에 의해 워핑이 수행된 후) 워핑 엔진 (622) 으로부터의 출력은 p_Y, p_U, p_V 로서 도 6에 나타낸 예측들을 포함하고, 여기서 p_Y 는 루마 성분 ($X_Y^{(t)}$)에 대한 예측에 대응하고, p_U 는 크로마 성분($X_U^{(t)}$)에 대한 예측에 대응하고, p_V 는 현재 프레임 (614) 의 크로마 성분($X_V^{(t)}$)에 대한 예측에 대응한다.

[0108] 그 다음, 딥 러닝 기반 시스템(600)은 루마 성분에 대한 레지듀얼 신호(r_Y), 크로마 성분($X_U^{(t)}$)에 대한 레지듀얼 신호(r_U), 및 크로마 성분($X_V^{(t)}$)에 대한 레지듀얼 신호(r_V)를 포함하는 레지듀얼 신호들을 획득하기 위해, 현재 프레임(614)의 대응하는 루마 성분($X_Y^{(t)}$) 및 크로마 성분($X_U^{(t)}$, $X_V^{(t)}$)으로부터 예측들(p_Y, p_U, p_V)을 감산할 수 있다. 레지듀얼 엔진(620)은 레지듀얼에 대한 잠재적 표현($\hat{\boldsymbol{y}}$)을 생성할 수 있다. 레지듀얼의 잠재적 표현($\hat{\boldsymbol{y}}$) (또는 다른 디바이스로부터 수신된 레지듀얼의 잠재적 표현)을 사용하여, 레지듀얼 엔진(620)은 루마 성분에 대한 재구성된 레지듀얼 신호($\hat{\mathrm{r}}_Y$), 크로마 성분($X_U^{(t)}$)에 대한 재구성된 레지듀얼 신호($\hat{\mathrm{r}}_U$), 및 크로마 성분($X_V^{(t)}$)에 대한 재구성된 레지듀얼 신호($\hat{\mathrm{r}}_V$)를 포함하는 현재 프레임에 대한 재구성된 레지듀얼을 생성할 수 있다. 딥 러닝 기반 시스템(600)은 재구성된 프레임(616)을 생성하기 위해, 재구성된 레지듀얼($\hat{\mathrm{r}}_Y$,

$\hat{r}_U$, $\hat{r}_V$)에 예측들(p_Y, p_U, p_V)을 추가할 수 있다.

[0109] 도 7a 는 루마 컴포넌트들 (722) 로서 집합적으로 도시된, (시간 t에서의) 현재 프레임의 루마 컴포넌트 ($X_Y^{(t)}$) 및 (시간 t-1에서의) 이전 프레임의 재구성된 루마 컴포넌트($\hat{X}_Y^{(t-1)}$)로 동작하는 플로우 엔진 (718) 의 일 예를 예시하는 다이어그램이다. 전술한 바와 같이, 일부 경우에 플로우 엔진(718)은 오토인코더(VAE_{flow})로서 구현될 수 있다. 일부 경우들에서, 결합된 딥 러닝-기반 아키텍처는, 도 7a에 도시된 바와 같이, 플로우 엔진(718)이 루마 모션 정보(예를 들어, SSF f^L) 및 크로마 모션 정보(예를 들어, SSF f^C)를 추정하기 위해 현재 프레임 ($X_Y^{(t)}$) 및 이전에 재구성된 프레임 ($\hat{X}_Y^{(t-1)}$) 둘 모두의 루마 성분을 사용하도록 설계될 수 있다. 예를 들어, 본원에 설명된 바와 같이, 크로마 모션 정보 (예를 들어, f^C) 는 루마 모션 정보 (예를 들어, f^L)에 기초하여 도출될 수 있다.

[0110] 도 7a에 도시된 바와 같이, 현재 프레임의 루마 컴포넌트($X_Y^{(t)}$)에 대한 모션 정보 (f^L)를 결정하기 위해, 현재 프레임의 루마 컴포넌트 ($X_Y^{(t)}$) 및 이전 프레임의 재구성된 루마 컴포넌트($\hat{X}_Y^{(t-1)}$)는 여러 컨볼루션 계층들 및 활성화 계층들에 의해 프로세싱된다 (집합적으로 순방향 패스 (723) 로서 도시됨). 도 7a의 "↓2" 및 "↑2" 표기는 스트라이드(stride) 값들을 지칭하고, 여기서 ↓2는 (↓"에 의해 표시된 바와 같은 다운샘플링에 대한) 2의 스트라이드를 지칭하고, ↑2는 또한 (↑"에 의해 표시된 바와 같은 업샘플링에 대한) 2의 스트라이드를 지칭한다. 예를 들어, 컨볼루션 계층 (724) 은 2의 스트라이드 값만큼 수평 및 수직 차원들에서 5x5 컨볼루션 필터를 적용함으로써 4 의 팩터만큼 입력 루마 성분들($X_Y^{(t)}$ 및 $\hat{X}_Y^{(t-1)}$)을 다운샘플링한다. 컨볼루션 계층 (724) 의 결과적인 출력은 현재 프레임의 루마 성분($X_Y^{(t)}$)에 대한 루마 모션 정보 (f^L) 를 나타내는 피처 값들의 N 개의 어레이들 (N 개의 채널들에 대응함) 이다. 표기 "2/N"은 2개의 입력 채널 및 N개의 출력 채널을 나타낸다. 컨볼루션 계층(724)을 뒤따르는 비선형 계층은 컨볼루션 계층(724)에 의해 출력된 특징 값들을 프로세싱할 수 있다. 연속적인 컨볼루션 계층들 및 비선형 계층들 각각은 순방향 패스(723)의 최종 컨볼루션 계층(725)이 플로우 엔진(718)의 병목 부분(726)에 피처들을 출력할 때까지 이전 계층에 의해 출력된 피처들을 처리할 수 있다.

[0111] 순방향 패스(723)의 출력은 플로우 엔진(718)의 병목 부분(726)에 의해 처리되어 현재 프레임의 루마 성분 ($X_Y^{(t)}$)에 대한 루마 모션 정보(f^L)를 나타내는 비트스트림 또는 레이턴트를 생성한다. 병목 부분(726)은 순방향 패스(723)에서의 양자화 엔진 및 엔트로피 인코딩 엔진, 및 플로우 엔진(718)의 역방향 패스(728)에서의 엔트로피 디코딩 엔진 및 역양자화 엔진을 포함할 수 있다. 예를 들어, 양자화 엔진은 순방향 패스(723)의 최종 컨볼루션 계층(725)에 의해 출력된 피처들에 대해 양자화를 수행하여 양자화된 출력을 생성할 수 있다. 엔트로피 인코딩 엔진은 양자화 엔진으로부터의 양자화된 출력을 엔트로피 인코딩하여 비트스트림을 생성할 수 있다. 일부 경우들에서, 엔트로피 인코딩 엔진은 엔트로피 인코딩을 수행하기 위해 하이퍼프라이어 네트워크에 의해 생성된 프리어를 사용할 수 있다. 뉴럴 네트워크 시스템은 저장을 위해, 다른 디바이스로의 송신을 위해, 서버 디바이스 또는 시스템에 비트스트림을 출력할 수 있고, 그리고/또는 그렇지 않으면 비트스트림을 출력할 수 있다.

[0112] 역방향 패스(728)는 일부 경우들에서, 플로우 엔진(718)의 뉴럴 네트워크 시스템의 디코더 서브-네트워크 또는 (다른 디바이스의) 다른 플로우 엔진의 뉴럴 네트워크 시스템의 디코더 서브-네트워크일 수 있다. 플로우 엔진(718)의 엔트로피 디코딩 엔진은 병목(726)의 엔트로피 인코딩 엔진(또는 다른 플로우 엔진의 병목의 엔트로피 인코딩 엔진)에 의해 출력된 비트스트림을 엔트로피 디코딩하고, 엔트로피 디코딩된 데이터를 역방향 패스(728)의 역양자화 엔진에 출력할 수 있다. 엔트로피 디코딩 엔진은 엔트로피 디코딩을 수행하기 위해 하이퍼프라이어 네트워크에 의해 생성된 프라이어를 사용할 수 있다. 역양자화 엔진은 데이터를 역양자화할 수 있다.

[0113] 역방향 패스 (728) 의 컨볼루션 계층들 및 역 활성화 계층들은 그 후 병목 (726) 으로부터의 역양자화된 데이터를 프로세싱하여 현재 프레임의 루마 성분($X_Y^{(t)}$)에 대한 모션 정보 (729) (f^L) 를 생성할 수도 있다. 모션 정보

(729) (f^L) 는 현재 프레임의 루마 성분($X_Y^{(t)}$)의 각각의 샘플에 대한 모션 벡터와 같은 모션 벡터들 (예를 들어, 수평 또는 x-방향으로의 크기 및 수직 또는 y-방향으로의 크기를 가짐) 을 포함할 수 있다. 일부 경우에, 모션 정보(729)(fL)는 스케일 성분을 더 포함할 수 있다. 예를 들어, 예시를 위해 도 7a에 도시된 바와 같이, 모션 정보 (729) 는 v_x^L 성분, v_y^L 성분, 및 s^L 성분을 포함한다. 위에서 언급된 바와 같이, v_x^L, v_y^L, 및 s^L 성분들은 예측들 p_Y, p_U, p_V을 생성하기 위해 (시간 t에서) 현재 프레임(614)의 픽셀들을 워핑하도록 워핑 엔진(622)에 의해 식 (1)에서 사용될 수 있다.

[0114] 현재 프레임의 루마 성분($X_Y^{(t)}$)에 대한 모션 정보(729)(f^L)를 학습한 후, 플로우 엔진(718)은 현재 프레임의 크로마 성분들에 대한 모션 정보(731)(f^C)를 결정 또는 예측할 수 있다. 예를 들어, 플로우 엔진 (718) 은 크로마 성분들에 대한 모션 정보 (731) (f^C) 를 획득하기 위해 루마 성분($X_Y^{(t)}$)에 대한 모션 정보 (729) (f^L) 를 서브샘플링할 수도 있다. 크로마 성분들에 대한 모션 정보 (731) (f^C) 는 현재 프레임의 크로마 성분들의 각각의 샘플에 대한 모션 벡터와 같은 모션 벡터들 (예를 들어, 수평 또는 x-방향으로의 크기 및 수직 또는 y-방향으로의 크기를 가짐) 을 포함할 수 있다. 일부 경우에, 모션 정보(731)(f^C)는 스케일 성분을 더 포함할 수 있다. 예를 들어, 도 7a에 도시된 바와 같이, 현재 프레임의 크로마 성분들에 대한 모션 정보(731)(f^C)는 v_x^C 성분, v_y^C 성분 및 s^C 성분을 포함한다. 루마 성분에 대한 모션 정보 (729) (f^L) 와 유사하게, 크로마 모션 정보 (731) (f^C)의 v_x^C, v_y^C 및 s^C 성분들은 식(1)에서 이용되어 워핑 엔진(622)에 의해 (시간 t 에서) 현재 프레임 (614) 의 픽셀들을 워핑하여 예측들 p_Y, p_U, p_V 을 생성할 수 있다.

[0115] 일부 양태들에서, 다운 샘플링을 갖는 컨볼루션 계층 (730) 은 현재 프레임의 루마 성분($X_Y^{(t)}$)에 대한 모션 정보 (729) (fL)에 기초하여 현재 프레임의 크로마 성분들에 대한 모션 정보 (731) (fC) 를 학습하도록 (예를 들어, 비지도 학습 또는 트레이닝을 사용하여) 트레이닝될 수 있다. 하나의 예시적인 예에서, 플로우 엔진(718)을 트레이닝하기 위해 사용될 수 있는 트레이닝 세트는 루마 및 크로마 모션 정보를 (실측 정보로서) 포함할 수 있다. 루마 모션 정보는 플로우 엔진(718)의 뉴럴 네트워크에 입력될 수 있고, 플로우 엔진(718)으로부터 출력되는 결과적인 크로마 모션 정보는 손실 함수(예를 들어, L1 또는 절대 차이들의 합, L2 놈 또는 제곱 차이들의 합, 또는 다른 손실 함수)를 사용하여 실측 크로마 모션 정보를 사용하여 최소화될 수 있다.

[0116] 컨볼루션 계층(730)은 도 7a에서 |3/3|5x5 conv↓2| 로서 표시된다. 표기 "3/3"은 3개의 출력 채널을 초래하는 3개의 입력 채널이 있음을 나타낸다. 위에서 언급된 바와 같이, "↓2" 및 "↑2" 표기는 스트라이드 값들을 지칭하며, ↓2는 ("↓"에 의해 표시된 바와 같은) 다운샘플링에 대한 2의 스트라이드를 지칭하고, ↑2는 ("↑"에 의해 표시된 바와 같은) 업샘플링에 대한 2의 스트라이드를 지칭한다. 예를 들어, 컨볼루션 계층(730)은 2의 스트라이드 값만큼 수평 및 수직 차원에서 5x5 컨볼루션 필터를 적용함으로써 루마 성분($X_Y^{(t)}$)에 대한 모션 정보(729)(f^L)를 4의 팩터만큼 (예를 들어, YUV 4:2:0 포맷에 대해) 다운샘플링한다. 일부 예들에서, 컨볼루션 계층(730)은 다른 포맷들(예를 들어, YUV 4:2:2 포맷 등)에 대한 다른 인자들에 의해 다운샘플링하도록 트레이닝될 수 있다. 컨볼루션 계층(724)의 결과적인 출력은 루마 성분($X_Y^{(t)}$)에 대한 모션 정보(729)(f^L)의 다운샘플링된 버전인 피처 값들의 3x3 어레이(3개의 채널들에 대응함)이다.

[0117] (도 7a에 도시되지 않은) 다른 양태들에서, 현재 프레임의 크로마 성분에 대한 모션 정보 (731) (fC) 는 루마 성분($X_Y^{(t)}$)에 대한 모션 정보 (729) (f^L) 를 직접 서브샘플링함으로써 획득될 수 있다. 예를 들어, 플로우 엔진 (718) 은 루마 플로우(들)를 프로세싱하기 위해 컨볼루션 계층 (730) 을 사용하지 않고서 크로마 플로우를 결정할 수 있다. 하나의 예시적인 예에서, 컨볼루션 계층 (730) 대신에, 플로우 엔진 (718) 은 크로마 모션 정보 (731) (f^C) 를 획득하기 위해 루마 모션 정보 (729) (f^L) 를 직접 서브샘플링할 수 있는 서브샘플러 (플로우 엔진 (718) 의 뉴럴 네트워크로부터 분리될 수 있음) 를 포함할 수 있다.

[0118] 도 7b 는 현재 프레임에 대한 크로마 모션 정보를 획득하기 위해 (예를 들어, 도 7a 의 플로우 엔진 (718) 을 사용하여) 현재 프레임에 대해 결정된 루마 모션 정보를 서브샘플링하기 위한 서브샘플링 엔진 (735) 의 일 예를 예시한 다이어그램이다. 예시의 목적으로, 단순화된 예에는 4x4 (4 개의 행들 및 4 개의 열들) 의 해상도를 갖는 루마 모션 정보 (732) 의 각각의 채널 N (N = 2) 이 제공되며, 총 16 개의 플로우 모션 또는 변위 벡터들을 갖는다. 서브샘플링 엔진(735)은 루마 모션 정보(732)의 서브샘플링된/다운샘플링된 버전인 크로마 모션 정보(738)를 생성 또는 획득하기 위해 루마 모션 정보(732)를 서브샘플링 또는 다운샘플링한다.

[0119] 도 7b 의 예시적인 예는 루마 모션 정보 (732) 의 사이즈의 쿼터인 크로마 모션 정보 (738) 를 도시한다. 예를 들어, 이전에 설명된 바와 같이, YUV 4:2:0 포맷을 갖는 콘텐츠에 대해, U 및 V 채널 해상도는 Y 채널 해상도의 절반이다(U 및 V 채널들은 폭 및 높이가 절반으로 되기 때문에, Y 채널의 4분의 1인 크기를 갖는다). 서브샘플링 엔진(735)은 4:2:0 포맷 이외의 다른 포맷들을 처리하도록 트레이닝되거나 달리 구성될 수 있으며, 이 경우 서브샘플링은 도 7a에 도시된 것과 상이한 해상도들을 갖는 크로마 정보를 생성하는 것을 포함할 수 있다.

[0120] 일부 양태들에서, 전술된 바와 같이, 서브샘플링 엔진(735)은 루마 모션 정보(732)로부터 크로마 모션 정보(738)를 결정하기 위해 (예를 들어, 비지도 학습 또는 트레이닝을 사용하여) 트레이닝될 수 있는 도 7a의 컨볼루션 계층(730)을 포함할 수 있다. 다른 양태들에서, 서브샘플링 엔진 (735) 은 크로마 모션 정보 (738) 를 획득하기 위해 루마 모션 정보 (732) 를 직접 서브샘플링하는 서브샘플러를 포함할 수 있다.

[0121] 순방향 패스(723) 및 역방향 패스(728)의 컨볼루션 또는 변환 계층들에서 뿐만 아니라 병목(M)에서의 채널들의 수(도 7a에서 N으로 표시됨)는 임의의 적절한 값으로 설정될 수 있다. 하나의 예시적인 예에서, 채널들의 수 N은 N=192 및 M=128로서 선택될 수 있다. (스케일 필드 s 와 연관된) 재구성된 프레임들의 연속적인 평활화된 버전들은 필터링 또는 평활화 연산자를 사용함으로써 획득될 수 있다. 일 예에서, 상이한 폭들을 갖는 가우시안 블러링 필터가 사용될 수 있다. 다른 예에서, 연속적인 필터링 및 보간을 갖는 가우시안 피라미드가 재구성된 프레임들의 평활화된 버전들을 생성하는 데 사용될 수 있다. 또한, 임의로 많은 수의 스케일들(S)이 사용될 수 있다. 일 예에서, 스케일 S는 S=3으로 설정될 수 있고, 스케일 레벨은 $s = [0, \sigma_0^2, \sigma_0^2 + (2\sigma_0)^2]$ 로서 선택될 수 있고, 여기서 σ_0 는 가우시안 필터 폭을 나타낼 수 있다.

[0122] 도 7a의 비선형 활성화 계층들은 예시적인 목적들을 위해 PReLU들로서 도시되지만, 일반화된 분할 정규화(generalized divisive normalization; GDN) 계층들, PReLU 및 GDN 계층들의 조합과 같은 다른 타입들의 비선형 활성화 계층들이 사용될 수 있다.

[0123] 일부 예들에서, 하나 이상의 YUV 포맷들(예를 들어, YUV 4:2:0)을 효과적으로 지원하기 위해, 도 6의 인트라 예측 엔진(602) 및 레지듀얼 엔진(620)은 도 8a, 도 9 또는 도 10에 도시된 일반적인 뉴럴 네트워크 아키텍처들에 기초하여 설계될 수 있다. 예를 들어, 도 8a, 도 9 및 도 10에 도시된 아키텍처들은 YUV 4:2:0 포맷을 갖는 입력 데이터를 처리하도록 구성될 수 있다. 일부 예들에서, 도 8a, 도 9 또는 도 10에 도시된 것과 유사한 뉴럴 네트워크 아키텍처는 다른 유형의 YUV 콘텐츠(예를 들어, YUV 4:4:4 포맷, YUV 4:2:2 포맷 등을 갖는 콘텐츠) 및/또는 다른 입력 포맷을 갖는 콘텐츠를 인코딩 및/또는 디코딩하는데 사용될 수 있다. 일부 경우들에서, 도 8a, 도 9 및 도 10 에 도시된 각각의 아키텍처는 YUV(예를 들어, 4:2:0) 레지듀얼로 동작하는 레지듀얼 오토인코더를 포함한다.

[0124] 도 8a는 4:2:0 입력 (Y, U 및 V) 데이터와 직접 동작하도록 구성될 수 있는 프론트-엔드 뉴럴 네트워크 시스템(800)의 예를 예시하는 도면이다. 도 8a에 도시된 바와 같이, 뉴럴 네트워크 시스템의 인코더 서브-네트워크(순방향 패스로 또한 지칭됨)에서, 분지형 루마 및 크로마 채널들(루마 Y 채널(802) 및 U 및 V 크로마 채널들(804))은 1x1 컨볼루션 계층(806)을 사용하여 결합된 다음, 비선형 계층(808)(비선형 연산자로 또한 지칭됨)이 적용된다. 유사한 동작들이 뉴럴 네트워크 시스템의 디코더 서브-네트워크(역방향 패스라고도 함)에서 수행되지만, 역순으로 수행된다. 예를 들어, 도 8a에 도시된 바와 같이, 역 비선형 계층(809)(역 비선형 연산자로도 지칭됨)이 적용되고, Y 및 U, V 채널들은 1x1 컨볼루션 계층(813)을 사용하여 분리되고, 개별 Y 및 U, V 채널들은 각각의 역 비선형 계층들(815, 816) 및 컨볼루션 계층들(817, 818)을 사용하여 프로세싱된다.

[0125] 도 8a의 뉴럴 네트워크 시스템(800)의 인코더 서브-네트워크의 첫 번째 2개의 뉴럴 네트워크 계층들은 제1 컨볼루션 계층(811)(Nconv |3x3|↓1로 표시됨), 제2 컨볼루션 계층(810)(Nconv |5x5|↓2로 표시됨), 제1 비선형 계층(814) 및 제2 비선형 계층(812)을 포함한다. 도 8a의 프론트-엔드 뉴럴 네트워크 아키텍처의 디코더 서브-네트워크에서의 마지막 2개의 뉴럴 네트워크 계층들은 제1 역 비선형 계층(816), 제2 역 비선형 계층(815), 프레임의 재구성된 색차 (U 및 V) 성분들을 생성하기 위한 제1 컨볼루션 계층(818)(2conv |3x3|↑1로 표시됨), 및

프레임의 재구성된 휘도 (Y) 성분을 생성하기 위한 제2 컨볼루션 계층(817)(1conv |5x5|↑2로 표시됨)을 포함한다. "Nconv" 표기는 주어진 컨볼루션 계층의 출력 채널들의 수(N)(출력 특징들의 수에 대응함)를 지칭한다(N의 값은 출력 채널들의 수를 정의한다). 3x3 및 5x5 표기는 각각의 컨볼루션 커널들(예를 들어, 3x3 커널 및 5x5 커널)의 크기를 나타낸다. "↓1" 및 "↓2" 표기는 스트라이드(stride) 값들을 지칭하고, 여기서 ↓1은 (↓"에 의해 표시된 바와 같은 다운샘플링에 대한) 1 의 스트라이드를 지칭하고, ↓2는 (다운샘플링에 대한) 1 의 스트라이드를 지칭한다. "↑1" 및 "↑2" 표기는 스트라이드 값들을 지칭하고, 여기서 ↑1은 (↑"에 의해 표시된 바와 같은 업샘플링에 대한) 1 의 스트라이드를 지칭하고, ↑2는 (업샘플링에 대한) 1 의 스트라이드를 지칭한다.

[0126] 예를 들어, 컨볼루션 계층 (810) 은 2 의 스트라이드 값만큼 수평 및 수직 차원들에서 5x5 컨볼루션 필터를 적용함으로써 4 의 팩터만큼 입력 루마 채널 (802) 을 다운샘플링한다. 컨볼루션 계층(810)의 결과적인 출력은 피처 값들의 N개의 어레이들(N개의 채널들에 대응함)이다. 컨볼루션 계층(811)은 1의 스트라이드 값에 의해 수평 및 수직 차원에서 3x3 컨볼루션 필터를 적용함으로써 입력 크로마(U 및 V) 채널(804)을 프로세싱한다. 컨볼루션 계층(811)의 결과적인 출력은 피처 값들의 N개의 어레이들(N개의 채널들에 대응함)이다. 컨볼루션 계층(810)에 의해 출력된 피처 값들의 어레이들은 컨볼루션 계층(811)에 의해 출력된 피처 값들의 어레이들과 동일한 치수를 갖는다. 비선형 계층(812)은 그 후 컨볼루션 계층(810)에 의해 출력된 피처 값들을 처리할 수 있고, 비선형 계층(814)은 컨볼루션 계층(811)에 의해 출력된 피처 값들을 처리할 수 있다.

[0127] 그 후, 1x1 컨볼루션 계층(806)은 비선형 계층들(812, 814)에 의해 출력된 피처 값들을 프로세싱할 수 있다. 1x1 컨볼루션 계층(806)은 루마 채널(802) 및 크로마 채널들(804)과 연관된 특징들의 선형 조합을 생성할 수 있다. 선형 조합 연산은 Y 및 UV 성분들의 값 당 크로스-채널 혼합(per-value cross-channel mixing)으로서 동작하여, 코딩 성능을 향상시키는 크로스-성분(예를 들어, 크로스-휘도 및 색차 성분) 예측을 초래한다. 1x1 컨볼루션 계층(806)의 각각의 1x1 컨볼루션 필터는 루마 채널(802)의 대응하는 N번째 채널 및 크로마 채널들(804)의 대응하는 N번째 채널에 적용되는 각각의 스케일링 인자를 포함할 수 있다.

[0128] 도 8b는 1x1 컨볼루션 계층(838)의 예시적인 동작을 예시하는 도면이다. 전술한 바와 같이, N은 출력 채널의 수를 나타낸다. 도 8b 에 도시된 바와 같이, N-채널 크로마(결합된 U 및 V) 출력(832) 및 N-채널 루마(Y) 출력(834)을 포함하는 2N 채널들이 1x1 컨볼루션 계층(838)에 대한 입력으로서 제공된다. 도 8b 의 예에서, N 의 값은 2 와 동일하여, N-채널 크로마 출력 (832)에 대한 값들의 2 개의 채널들 및 N-채널 루마 출력 (834)에 대한 값들의 2 개의 채널들을 표시한다. 도 8a 를 참조하면, N-채널 크로마 출력 (832) 은 비선형 계층 (814) 으로부터의 출력일 수 있고, N-채널 루마 출력 (834) 은 비선형 계층 (812) 으로부터의 출력일 수 있다.

[0129] 1x1 컨볼루션 계층(838)은 2N 채널들을 처리하고 2N 채널들의 특징적 선형 결합을 수행한 다음, 특징들 또는 계수들의 N-채널 세트를 출력한다. 1x1 컨볼루션 계층(838)은 2개의 1x1 컨볼루션 필터들(N=2에 기초함)을 포함한다. 제 1 1x1 컨볼루션 필터는 s1 값으로 도시되고, 제 2 1x1 컨볼루션 필터는 s2 값으로 도시된다. s1 값은 제 1 스케일링 인자를 나타내고 s2 값은 제 2 스케일링 인자를 나타낸다. 하나의 예시적인 예에서, s1 값은 3 과 동일하고 s2 값은 4 와 동일하다. 1x1 컨볼루션 계층(838)의 1x1 컨볼루션 필터들 각각은 1의 스트라이드 값을 가지며, 이는 스케일링 인자들 s1 및 s2 가 UV 출력(832) 및 Y 출력(834)에서의 각각의 값에 적용될 것임을 나타낸다.

[0130] 예를 들어, 제 1 1x1 컨볼루션 필터의 스케일링 인자 s1 은 UV 출력(832)의 제 1 채널(C1) 내의 각각의 값 및 Y 출력(834)의 제 1 채널(C1) 내의 각각의 값에 적용된다. UV 출력(832)의 제 1 채널(C1)의 각각의 값 및 Y 출력(834)의 제 1 채널(C1)의 각각의 값이 제 1 1x1 컨볼루션 필터의 스케일링 인자 s1 에 의해 스케일링되면, 스케일링된 값들은 출력 값들(839)의 제 1 채널(C1)로 결합된다. 제 2 1x1 컨볼루션 필터의 스케일링 인자 s2 는 UV 출력(832)의 제 2 채널(C2) 내의 각각의 값 및 Y 출력(834)의 제 2 채널(C2) 내의 각각의 값에 적용된다. UV 출력 (832) 의 제 2 채널 (C2) 의 각각의 값 및 Y 출력 (834) 의 제 2 채널 (C2) 의 각각의 값이 제 2 1x1 컨볼루션 필터의 스케일링 팩터 s2 만큼 스케일링된 후, 스케일링된 값들은 출력 값들 (839) 의 제 2 채널 (C2)에 결합된다. 그 결과, 4개의 Y 및 UV 채널들(2개의 Y 채널들 및 2개의 결합된 UV 채널들)은 2개의 출력 채널들(C1 및 C2)로 혼합되거나 결합된다.

[0131] 도 8a로 돌아가면, 1x1 컨볼루션 계층(806)의 출력은 인코더 서브-네트워크의 추가적인 비선형 계층들 및 추가적인 컨볼루션 계층들에 의해 프로세싱된다. 병목(820)은 인코더 서브-네트워크(또는 순방향 패스) 상의 양자화 엔진 및 엔트로피 인코딩 엔진 및 디코더 서브-네트워크(또는 역방향 패스) 상의 엔트로피 디코딩 엔진 및 역양자화 엔진을 포함할 수 있다. 양자화 엔진은 인코더 서브-네트워크의 최종 뉴럴 네트워크 계층(819)에 의해 출력된 특징들에 대해 양자화를 수행하여 양자화된 출력을 생성할 수 있다. 엔트로피 인코딩 엔진은 양자화

엔진으로부터의 양자화된 출력을 엔트로피 인코딩하여 비트스트림을 생성할 수 있다. 일부 경우들에서, 엔트로피 인코딩 엔진은 엔트로피 인코딩을 수행하기 위해 하이퍼프라이어 네트워크에 의해 생성된 프리어를 사용할 수 있다. 뉴럴 네트워크 시스템은 저장을 위해, 다른 디바이스로의 송신을 위해, 서버 디바이스 또는 시스템에 비트스트림을 출력할 수 있고, 그리고/또는 그렇지 않으면 비트스트림을 출력할 수 있다.

[0132] 뉴럴 네트워크 시스템의 디코더 서브-네트워크 또는 (다른 디바이스의) 다른 뉴럴 네트워크 시스템의 디코더 서브-네트워크는 비트스트림을 디코딩할 수 있다. (디코더 서브-네트워크의) 병목(820)의 엔트로피 디코딩 엔진은 비트스트림을 엔트로피 디코딩하고 엔트로피 디코딩된 데이터를 디코더 서브-네트워크의 역양자화 엔진에 출력할 수 있다. 엔트로피 디코딩 엔진은 엔트로피 디코딩을 수행하기 위해 하이퍼프라이어 네트워크에 의해 생성된 프라이어를 사용할 수 있다. 역양자화 엔진은 데이터를 역양자화할 수 있다. 역양자화된 데이터는 디코더 서브-네트워크의 다수의 컨볼루션 계층들 및 다수의 역 비선형 계층들에 의해 프로세싱될 수 있다.

[0133] 여러 개의 컨볼루션 및 비선형 계층들에 의해 처리된 후, 1x1 컨볼루션 계층(813)은 최종 역 비선형 계층(809)에 의해 출력된 데이터를 처리할 수 있다. 1x1 컨볼루션 계층(813)은 데이터를 Y 채널 특징들 및 결합된 UV 채널 특징들로 분할할 수 있는 2N 컨볼루션 필터들을 포함할 수 있다. 예를 들어, 역 비선형 계층(809)에 의해 출력된 N개의 채널들 각각은 1x1 컨볼루션 계층(813)의 2N개의 1x1 컨볼루션들을 사용하여 처리될 수 있다(스케일링을 초래함). N 개의 입력 채널들에 적용되는 (총 2N 개의 출력 채널들로부터의) 출력 채널에 대응하는 각각의 스케일링 팩터 ni 에 대해, 디코더 서브-네트워크는 N 개의 입력 채널들에 걸쳐 합산을 수행하여, 2N 개의 출력들을 초래할 수 있다. 하나의 예시적인 예에서, 스케일링 팩터 n1 에 대해, 디코더 서브-네트워크는 스케일링 팩터 n1 을 N 개의 입력 채널들에 적용할 수 있고 그 결과를 합산할 수 있고, 이는 하나의 출력 채널을 초래한다. 디코더 서브-네트워크는 2N 개의 상이한 스케일링 팩터들 (예를 들어, 스케일링 팩터 n1, 스케일링 팩터 n2 내지 스케일링 팩터 n2N)에 대해 이 동작을 수행할 수 있다.

[0134] 1x1 컨볼루션 계층(813)에 의해 출력된 Y 채널 피처들은 역 비선형(815)에 의해 처리될 수 있다. 1x1 컨볼루션 계층(813)에 의해 출력된 결합된 UV 채널 피처들은 역 비선형 (816)에 의해 프로세싱될 수 있다. 컨볼루셔널 계층 (817) 은 Y 채널 피처들을 프로세싱하고 재구성된 Y 성분 (824) 으로서 도시된 재구성된 프레임의 샘플 또는 픽셀 (예를 들어, 휘도 샘플들 또는 픽셀들) 당 재구성된 Y 채널을 출력할 수 있다. 컨볼루션 계층(818)은 결합된 UV 채널 피처들을 프로세싱할 수 있고, 재구성된 U 및 V 성분들(825)로서 도시된, 재구성된 프레임의 픽셀 또는 샘플(예를 들어, 색차-블루 샘플들 또는 픽셀들) 당 재구성된 U 채널 및 재구성된 프레임의 픽셀 또는 샘플(예를 들어, 색차-레드 샘플들 또는 픽셀들) 당 재구성된 V 채널을 출력할 수 있다.

[0135] 일부 예들에서, 상이한 비선형성 연산자들을 갖는 도 8a의 아키텍처의 상이한 변형들이 인트라 예측 엔진(602) 및 레지듀얼 엔진(620)으로서 사용될 수 있다. 예를 들어, 도 9 및 도 10은 YUV 포맷을 갖는 데이터(예를 들어, Y, U 및 V 성분을 갖는 YUV 4:2:0 입력 데이터)를 처리하기 위해 구성된 도 8a의 프론트-엔드 아키텍처를 설명하기 위한 도면들이다. 도 9의 뉴럴 네트워크 시스템(900)에서, 인코더 측에서, 분지형 루마 및 크로마 채널들은 (도 8a의 것과 유사한) 1x1 컨볼루션 계층을 사용하여 결합되고, 그 후 GDN 비선형 연산자가 적용된다. 도 10의 뉴럴 네트워크 시스템(1000)에서, 인코더 측에서, 분지형 루마 및 크로마 채널들은 (도 8a의 것과 유사한) 1x1 컨볼루션 계층을 사용하여 결합되고, 그 후 PReLU 비선형 연산자가 적용된다. 일 예에서, VAE_{res} 및 VAE_{intra} 모두는 도 9에 도시된 변형을 사용할 수 있다. 다른 예에서, VAE_{res} 및 VAE_{intra} 모두는 도 10의 변형을 사용할 수 있다. 다른 예에서, VAE_{res} 는 도 9의 변형을 사용할 수 있고, VAE_{intra} 는 도 10의 변형을 사용할 수 있다. 다른 예에서, VAE_{intra} 는 도 9의 변형을 사용할 수 있고, VAE_{res} 는 도 10의 변형을 사용할 수 있다.

[0136] 도 11 은 비디오 데이터를 프로세싱하기 위한 프로세스 (1100) 의 일 예를 예시하는 플로우 다이어그램이다. 블록(1102)에서, 프로세스(1100)는 기계 학습 시스템에 의해, 입력 비디오 데이터를 획득하는 단계를 포함한다. 입력 비디오 데이터는 현재 프레임에 대한 적어도 하나의 휘도 성분 (예를 들어, 도 7a의 (시간 t에서의) 현재 프레임의 루마 성분 $X_Y^{(t)}$) 을 포함한다. 일부 경우들에서, 입력 비디오 데이터는, 적어도 하나의 재구성된 휘도 성분으로 지칭될 수 있는, 이전에 재구성된 프레임에 대한 적어도 하나의 휘도 성분 (예를 들어, 도 7a의 (시간 t-1에서의) 이전 프레임의 재구성된 루마 성분 $\hat{X}_Y^{(t-1)}$)을 포함한다. 일부 양태들에서, 현재 프레임은 비디오 프레임을 포함한다. 일부 경우들에서, 하나 이상의 색차 성분들은 적어도 하나의 색차-청색 성분 및 적어도 하나의 색차-적색 성분을 포함한다. 일부 양태들에서, 현재 프레임은 휘도-색차 (YUV) 포맷을 갖는다. 일부 경우에, YUV 포맷은 YUV 4:2:0 포맷이다.

[0137] 블록 1104 에서, 프로세스는 기계 학습 시스템에 의해, 현재 프레임에 대한 적어도 하나의 휘도 성분을 사용하여 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보 및 현재 프레임의 하나 이상의 색차 성분에 대한 모션 정보를 결정하는 단계를 포함한다. 일부 양태들에서, 프로세스 (1100) 는 현재 프레임의 적어도 하나의 휘도 성분 및 이전 프레임의 적어도 하나의 재구성된 루마 성분에 기초하여 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보를 결정하는 단계를 포함할 수도 있다. 일부 경우들에서, 프로세스 (1100) 는 현재 프레임의 적어도 하나의 휘도 성분에 대해 결정된 모션 정보를 사용하여 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하는 것을 더 포함할 수도 있다. 일부 경우들에서, 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보는 기계 학습 시스템의 컨볼루션 계층을 사용하여 결정된다. 예를 들어, 도 7a 를 예시적인 예로서 참조하면, 플로우 엔진 (718) 은 현재 프레임 $X_Y^{(t)}$ 및 이전에 재구성된 프레임 $\hat{X}_Y^{(t-1)}$ 양자 모두의 루마 성분을 사용하여 현재 프레임 $X_Y^{(t)}$ 에 대한 루마 모션 정보 (예를 들어, SSF f^L) 및 크로마 모션 정보 (예를 들어, SSF f^C) 를 추정할 수도 있다. 위에서 언급된 바와 같이, 크로마 모션 정보 (예를 들어, f^C) (731) 는 컨볼루션 계층 (730) 을 사용하여 루마 모션 정보 (예를 들어, f^L) (729)에 기초하여 도출될 수 있다. 일부 경우들에서, 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보는 현재 프레임의 적어도 하나의 휘도 성분에 대해 결정된 모션 정보를 샘플링함으로써 적어도 부분적으로 결정된다.

[0138] 일부 양태들에서, 프로세스(1100)는, 기계 학습 시스템에 의해, 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보 및 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 사용하여, 현재 프레임의 적어도 하나의 휘도 성분에 대한 워핑 파라미터 및 현재 프레임의 하나 이상의 색차 성분들에 대한 하나 이상의 워핑 파라미터들을 결정하는 것을 포함한다. 일부 양태들에서, 현재 프레임의 적어도 하나의 휘도 성분에 대한 워핑 파라미터 및 현재 프레임의 하나 이상의 색차 성분들에 대한 하나 이상의 워핑 파라미터들은 공간-스케일 플로우 (SSF) 워핑 파라미터들을 포함한다. 일부 경우들에서, 공간-스케일 플로우(SSF) 워핑 파라미터들은 학습된 스케일-흐름 벡터들을 포함한다. 예시적인 예로서 도 6을 참조하면, 워핑 파라미터들은 모션 또는 변위 벡터의 (x-방향으로의) 수평 성분을 표현하는 v_x^*, 모션 또는 변위 벡터의 (y-방향으로의) 수직 성분을 표현하는 v_y^*, 공간 모션/변위 정보 (vx 및 vy) 와 결합되는 재구성된 프레임들의 점진적으로 평활화된 버전을 표현하는 (스케일 필드로 지칭되는) s 를 포함할 수도 있다.

[0139] 프로세스(1100)는 현재 프레임의 적어도 하나의 휘도 성분에 대한 워핑 파라미터 및 현재 프레임의 하나 이상의 색차 성분들에 대한 하나 이상의 워핑 파라미터들을 사용하여 현재 프레임에 대한 하나 이상의 인터-프레임 예측들(예를 들어, 도 6의 예측자들 $\boldsymbol{p_V}$, $\boldsymbol{p_U}$, 및 $\boldsymbol{p_V}$)을 결정하는 것을 더 포함할 수 있다. 일부 경우들에서, 하나 이상의 인터-프레임 예측들은 현재 프레임의 적어도 하나의 휘도 성분에 대한 워핑 파라미터 및 현재 프레임의 하나 이상의 색차 성분들에 대한 하나 이상의 워핑 파라미터들을 사용하여 보간 연산을 적용함으로써 적어도 부분적으로 결정된다. 하나의 예시적인 예에서, 보간 연산은 삼선형 보간 연산을 포함한다.

[0140] 일부 예들에서, 본 명세서에 설명된 프로세스들은 도 11에 도시된 컴퓨팅 디바이스 아키텍처(1200)를 갖는 컴퓨팅 디바이스와 같은 컴퓨팅 디바이스 또는 장치에 의해 수행될 수 있다. 일 예에서, 프로세스 또는 프로세스들은 도 6에 도시된 뉴럴 네트워크 아키텍처 및/또는 도 7a, 도 7b, 도 8a, 도 9 및/또는 도 10에 도시된 뉴럴 네트워크 아키텍처들 중 임의의 하나 이상을 구현하는 컴퓨팅 디바이스 아키텍처(1200)를 갖는 컴퓨팅 디바이스에 의해 수행될 수 있다. 일부 예들에서, 컴퓨팅 디바이스는 모바일 디바이스(예를 들어, 모바일 폰, 태블릿 컴퓨팅 디바이스 등), 웨어러블 디바이스, 확장 현실 디바이스(예를 들어, 가상 현실(VR) 디바이스, 증강 현실(AR) 디바이스, 또는 혼합 현실(MR) 디바이스), 개인용 컴퓨터, 랩톱 컴퓨터, 비디오 서버, 텔레비전, 차량(또는 차량의 컴퓨팅 디바이스), 로봇 디바이스, 및/또는 본 명세서에 설명된 프로세스들을 수행하기 위한 리소스 능력들을 갖는 임의의 다른 컴퓨팅 디바이스를 포함하거나 그 일부일 수 있다.

[0141] 일부 경우들에서, 컴퓨팅 디바이스 또는 장치는 본 명세서에 설명된 프로세스들의 단계들을 수행하도록 구성되는 하나 이상의 입력 디바이스들, 하나 이상의 출력 디바이스들, 하나 이상의 프로세서들, 하나 이상의 마이크로프로세서들, 하나 이상의 마이크로컴퓨터들, 하나 이상의 송신기들, 수신기들 또는 결합된 송신기-수신기들(예를 들어, 트랜시버들로 지칭됨), 하나 이상의 카메라들, 하나 이상의 센서들, 및/또는 다른 컴포넌트(들)와 같은 다양한 컴포넌트들을 포함할 수 있다. 일부 예들에 있어서, 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 타입의 데이터를 통신 및/또는 수신하도록 구성될 수도 있다.

[0142] 컴퓨팅 디바이스의 컴포넌트들은 회로부에서 구현될 수 있다. 예를 들어, 컴포넌트들은 본 명세서에서 설명된 다양한 동작들을 수행하기 위해, 하나 이상의 프로그래밍가능 전자 회로들 (예컨대, 마이크로프로세서들, 그래픽스 프로세싱 유닛들 (GPU들), 디지털 신호 프로세서들 (DSP들), 중앙 프로세싱 유닛들 (CPU들), 뉴럴 프로세싱 유닛들 (NPU들) 및/또는 다른 적합한 전자 회로들) 을 포함할 수 있는 전자 회로들 또는 다른 전자 하드웨어를 포함할 수 있고/있거나 이들을 사용하여 구현될 수 있고, 및/또는 컴퓨터 소프트웨어, 펌웨어, 또는 이들의 임의의 조합을 포함할 수 있고/있거나 이들을 사용하여 구현될 수 있다.

[0143] 본원에 설명된 프로세스들은 논리 흐름도로서 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 경우, 기재된 동작들을 수행하는 하나 이상의 컴퓨터 판독가능 저장 매체들 상에 저장된 컴퓨터 실행가능 명령들을 나타낸다. 일반적으로, 컴퓨터 실행가능 명령들은 특정한 기능들을 수행하거나 또는 특정한 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.

[0144] 추가적으로, 본 명세서에서 설명된 프로세스들은 실행가능 명령들로 구성된 하나 이상의 컴퓨터 시스템들의 제어 하에서 수행될 수도 있고, 집합적으로 하나 이상의 프로세서들 상에서 실행하는 코드 (예를 들어, 실행가능 명령들, 하나 이상의 컴퓨터 프로그램들, 또는 하나 이상의 애플리케이션들) 로서, 하드웨어에 의해, 또는 이들의 조합으로 구현될 수도 있다. 앞서 언급된 바와 같이, 코드는 컴퓨터 판독가능 또는 머신 판독가능 저장 매체 상에, 예를 들어, 하나 이상의 프로세서들에 의해 실행가능한 복수의 명령들을 포함하는 컴퓨터 프로그램의 형태로 저장될 수도 있다. 컴퓨터 판독가능 또는 머신 판독가능 저장 매체는 비일시적일 수도 있다.

[0145] 도 12 는 본 명세서에 설명된 다양한 기술을 구현할 수 있는 일 예의 컴퓨팅 디바이스의 예시적인 컴퓨팅 디바이스 아키텍처 (1200) 를 도시한다. 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), 개인용 컴퓨터, 랩톱 컴퓨터, 비디오 서버, 차량(또는 차량의 컴퓨팅 디바이스), 또는 다른 디바이스를 포함할 수 있다. 예를 들어, 컴퓨팅 디바이스 아키텍처(1200)는 도 6 의 시스템을 구현할 수 있다. 컴퓨팅 디바이스 아키텍처(1200)의 컴포넌트들은 버스와 같은 연결(1205)을 사용하여 서로 전기적으로 통신하는 것으로 도시된다. 예시적인 컴퓨팅 디바이스 아키텍처(1200)는 프로세싱 유닛(CPU 또는 프로세서)(1210), 및 판독 전용 메모리(ROM)(1220) 및 랜덤 액세스 메모리(RAM)(1225)와 같은 컴퓨팅 디바이스 메모리(1215)를 포함하는 다양한 컴퓨팅 디바이스 컴포넌트들을 프로세서(1210)에 결합하는 컴퓨팅 디바이스 연결부(1205)를 포함한다.

[0146] 컴퓨팅 디바이스 아키텍처(1200)는 프로세서(1210)와 직접 연결되거나, 그에 근접하거나, 또는 그의 일부로서 통합된 고속 메모리의 캐시를 포함할 수 있다. 컴퓨팅 디바이스 아키텍처(1200)는 프로세서(1210)에 의한 빠른 액세스를 위해 메모리(1215) 및/또는 저장 디바이스(1230)로부터 캐시(1212)로 데이터를 복사할 수 있다. 이러한 방식으로, 캐시는 데이터를 기다리는 동안 프로세서(1210) 지연들을 회피하는 성능 부스트를 제공할 수 있다. 이들 및 다른 모듈들은 다양한 액션들을 수행하도록 프로세서 (1210) 를 제어하거나 제어하도록 구성될 수 있다. 다른 컴퓨팅 디바이스 메모리(1215)가 또한 사용가능할 수도 있다. 메모리(1215)는 상이한 성능 특성을 갖는 다수의 상이한 유형의 메모리를 포함할 수 있다. 프로세서 (1210) 는, 임의의 범용 프로세서 및 프로세서 (1210) 를 제어하도록 구성된, 저장 디바이스(1230) 에 저장된, 서비스 1 (1232), 서비스 2 (1234), 및 서비스 3 (1236)과 같은, 하드웨어 또는 소프트웨어 서비스뿐만 아니라, 소프트웨어 명령들이 프로세서 설계에 통합되는 특수 목적 프로세서를 포함할 수 있다. 프로세서 (1210) 는 다중 코어 또는 프로세서, 버스, 메모리 컨트롤러, 캐시 등을 포함하는 독립형 시스템일 수도 있다. 다중 코어 프로세서는 대칭 또는 비대칭일 수도 있다.

[0147] 컴퓨팅 디바이스 아키텍처 (1200) 와의 사용자 상호작용을 가능하게 하기 위해, 입력 디바이스 (1245) 는 스피치를 위한 마이크로폰, 제스처 또는 그래픽 입력을 위한 터치 감지 스크린, 키보드, 마우스, 모션 입력, 스피치 등과 같은 임의의 수의 입력 메커니즘을 나타낼 수 있다. 출력 디바이스 (1235) 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 (1200) 와 통신하기 위해 여러 유형의 입력을 제공하게 할 수 있다. 통신 인터페이스 (1240) 는 일반적으로 사용자 입력 및 컴퓨팅

디바이스 출력을 제어하고 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.

[0148] 저장 디바이스 (1230) 는 비휘발성 메모리이고, 하드 디스크 또는 자기 카세트, 플래시 메모리 카드, 고체 상태 메모리 디바이스, 디지털 다기능 디스크, 카트리지, 랜덤 액세스 메모리들 (RAM들) (1225), 판독 전용 메모리 (ROM) (1220) 및 이들의 하이브리드와 같은, 컴퓨터에 의해 액세스가능한 데이터를 저장할 수 있는 다른 유형의 컴퓨터 판독가능 매체일 수 있다. 저장 디바이스(1230)는 프로세서(1210)를 제어하기 위한 서비스들(1232, 1234, 1236)을 포함할 수 있다. 다른 하드웨어 또는 소프트웨어 모듈들이 고려된다. 저장 디바이스(1230)는 컴퓨팅 디바이스 연결(1205)에 연결될 수 있다. 일 양태에서, 특정 기능을 수행하는 하드웨어 모듈은, 그 기능을 수행하기 위해, 프로세서 (1210), 커넥션 (1205), 출력 디바이스 (1235) 등과 같은 필요한 하드웨어 컴포넌트들과 관련하여 컴퓨터 판독가능 매체에 저장된 소프트웨어 컴포넌트를 포함할 수 있다.

[0149] 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.

[0150] 용어 "디바이스" 는 (일 스마트폰, 일 제어기, 일 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 다중의 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.

[0151] 구체적 상세들은 본원에 제공된 실형태들 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 방법의 단계들 또는 루틴들, 또는 하드웨어와 소프트웨어의 조합들을 포함하는 기능적 블록들을 포함하는 개별의 기능적 블록들을 포함하는 것으로서 제시될 수도 있다. 도면들에서 도시되고/거나 본원에 기술된 것들 이외의 추가적인 컴포넌트들이 사용될 수도 있다. 예를 들어, 회로들, 시스템들, 네트워크들, 프로세스들, 및 다른 컴포넌트들은 그 실시형태들을 불필요한 상세로 불명료하게 하지 않기 위해 블록도 형태의 컴포넌트들로서 도시될 수도 있다. 다른 예들에서, 잘 알려진 회로들, 프로세스들, 알고리즘들, 구조들, 및 기술들은, 실시형태들을 불명료하게 하는 것을 회피하기 위해 불필요한 상세 없이 도시될 수도 있다.

[0152] 개별 실시형태들은, 플로우차트, 흐름도, 데이터 흐름도, 구조도, 또는 블록도로서 도시되는 프로세스 또는 방법으로서 위에서 설명될 수도 있다. 비록 플로우차트가 동작들을 순차적인 프로세스로서 기술할 수도 있지만, 동작들 중 다수는 병렬로 또는 동시에 수행될 수 있다. 부가적으로, 동작들의 순서는 재배열될 수도 있다. 프로세스는, 그의 동작들이 완료될 때 종료되지만, 도면에 포함되지 않은 추가적인 단계들을 가질 수 있다. 프로세스는 방법, 함수, 절차, 서브루틴, 서브프로그램 등에 대응할 수도 있다. 프로세스가 함수에 대응할 경우, 그의 종료는 그 함수의 호출 함수 또는 메인 함수로의 복귀에 대응할 수 있다.

[0153] 상술된 예들에 따른 프로세스들 및 방법들은 컴퓨터 판독가능 매체들에 저장되거나 그 외에 컴퓨터 판독가능 매체들로부터 이용가능한 컴퓨터 실행가능 명령들을 이용하여 구현될 수 있다. 이러한 명령들은, 예를 들어, 범용 컴퓨터, 특수 목적 컴퓨터, 또는 프로세싱 디바이스가 특정 기능 또는 기능들의 그룹을 수행하게 하거나 그 외에 수행하도록 구성하는 명령들 및 데이터를 포함할 수 있다. 사용되는 컴퓨터 리소스들의 부분들은 네트워크를 통해 액세스가능할 수 있다. 컴퓨터 실행 가능 명령들은 예를 들어 바이너리, 어셈블리 언어, 펌웨어, 소스 코드 등과 같은 중간 형식 명령일 수도 있다.

[0154] 용어 "컴퓨터 판독가능 매체" 는, 휴대 또는 비휴대 저장 디바이스, 광학 저장 디바이스, 및 명령(들) 및/또는 데이터를 저장, 포함 또는 나를 수 있는 다양한 다른 매체를 포함하지만, 이에 한정되지는 않는다. 컴퓨터 판

독 가능 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 예들은, 특히 자기 디스크 또는 테이프, 플래시 메모리와 같은 광학 저장 매체, 메모리 또는 메모리 디바이스들, 자기 또는 광학 디스크들, 플래시 메모리, 비휘발성 메모리가 제공된 USB 디바이스들, 네트워크화된 저장 디바이스들, 컴팩트 디스크(CD) 또는 디지털 다기능 디스크(DVD), 또는 이들의 임의의 적절한 조합을 포함할 수 있지만, 이들로 제한되지 않는다. 컴퓨터 판독가능 매체는, 절차, 함수, 서브프로그램, 프로그램, 루틴, 서브루틴, 모듈, 소프트웨어 패키지, 클래스, 또는 명령들, 데이터 구조들, 또는 프로그램 스테이트먼트들의 임의의 조합을 나타낼 수도 있는 코드 및/또는 머신 실행가능 명령들이 저장될 수도 있다. 코드 세그먼트는, 정보, 데이터, 인수들 (arguments), 파라미터들, 또는 메모리 컨텐츠를 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 전송 등을 포함한 임의의 적합한 수단을 통해 전달, 포워딩, 또는 전송될 수도 있다.

[0155] 일부 실시형태들에서, 컴퓨터 판독가능 저장 디바이스들, 매체들, 및 메모리들은 비트 스트림 등을 포함하는 무선 신호 또는 케이블을 포함할 수 있다. 하지만, 언급될 때, 비일시적인 컴퓨터 판독가능 저장 매체들은 에너지, 캐리어 신호들, 전자기 파들, 및 신호들 그 자체와 같은 매체들을 명시적으로 배제한다.

[0156] 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 태스크들을 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 태스크들을 수행할 수도 있다. 폼 팩터들의 통상적인 예들은 랩탑들, 스마트 폰들, 모바일 폰들, 태블릿 디바이스들 또는 다른 소형 폼 팩터 개인용 컴퓨터들, 개인용 디지털 보조기들, 랙마운트 디바이스들, 자립형 디바이스들 등을 포함한다. 본 명세서에서 설명된 기능은 또한, 주변기기들 또는 애드-인 (add-in) 카드들에서 구현될 수 있다. 그러한 기능은 또한, 추가의 예에 의해, 단일 디바이스에서 실행되는 상이한 칩들 또는 상이한 프로세스들 중에서 회로 보드 상에서 구현될 수 있다.

[0157] 명령들, 이러한 명령들을 운반하기 위한 매체들, 그것들을 시행하기 위한 컴퓨팅 리소스들, 및 이러한 컴퓨팅 리소스들을 지원하기 위한 다른 구조들은 본 개시물에서 설명될 기능들을 제공하기 위한 예시적인 수단들이다.

[0158] 전술한 설명에서, 본 출원의 양태들은 그것들의 특정 실시형태들을 참조하여 설명되었지만, 당업자는 본원이 이에 제한되지 않는다는 것을 인식할 것이다. 따라서, 본 출원의 예시적인 실시형태들이 본원에 상세히 설명되었지만, 본 발명의 개념은 달리 다양하게 구체화되고 채택될 수 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 어플리케이션들에서 활용될 수 있다. 본 명세서 및 도면들은, 이에 따라, 제한적이라기 보다는 예시적인 것으로서 간주되어야 한다. 예시의 목적으로, 방법들은 특정 순서로 설명되었다. 대안적인 실시형태들에 있어서, 방법들은 설명된 것과는 상이한 순서로 수행될 수도 있음이 인식되어야 한다.

[0159] 당업자는 본 명세서에서 사용된 미만 ("<") 및 초과 (">") 기호들 또는 용어가 본 개시의 범위로부터 일탈함 없이, 각각 이하 ("≤") 및 이상 ("≥") 기호들로 대체될 수 있다는 것을 알 것이다.

[0160] 컴포넌트들이 특정 동작을 수행 "하도록 구성된" 것으로 기술되는 경우, 그러한 구성은 예를 들어, 전자 회로 또는 다른 하드웨어를 동작을 수행하도록 설계함으로써, 프로그래밍 가능한 전자 회로 (예를 들어, 마이크로 프로세서 또는 다른 적절한 전자 회로) 를 동작을 수행하도록 프로그래밍함으로써, 또는 이들의 임의의 조합으로써 달성될 수 있다.

[0161] 문구 "~ 에 커플링된 (coupled to)" 은 다른 컴포넌트에 직접적으로 또는 간접적으로 물리적으로 접속된 임의의 컴포넌트, 및/또는, 다른 컴포넌트와 직접적으로 또는 간접적으로 통신하는 (예컨대, 유선 또는 무선 접속, 및/또는 다른 적합한 통신 인터페이스를 통해 다른 컴포넌트에 접속된) 임의의 컴포넌트를 지칭한다.

[0162] 세트 "중 적어도 하나" 또는 세트 "중 하나 이상" 을 인용하는 청구항 언어 또는 다른 언어는 그 세트의 하나의 멤버 또는 그 세트의 다중의 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 언어는 A, B, 또는 A 및 B 를 의미한다.

다른 예에서, "A, B, 및 C 중 적어도 하나" 또는 "A, B, 또는 C 중 적어도 하나"를 인용하는 청구항 언어는 A, B, C, 또는 A 및 B, 또는 A 및 C, 또는 B 및 C, 또는 A 및 B 및 C 를 의미한다. 언어 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 언어는 A, B, 또는 A 및 B 를 의미할 수 있으며, A 및 B 의 세트에 열거되지 않은 항목들을 추가적으로 포함할 수 있다.

[0163] 본 명세서에 개시된 실시형태들과 관련하여 설명된 다양한 예시적인 논리 블록들, 모듈들, 회로들, 및 알고리즘 단계들은 전자 하드웨어, 컴퓨터 소프트웨어, 펌웨어, 또는 이들의 조합들로서 구현될 수도 있다. 하드웨어와 소프트웨어의 이러한 상호대체 가능성을 분명히 예시하기 위해, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회로들 및 단계들이 일반적으로 그들의 기능의 관점에서 상기 설명되었다. 이러한 기능성이 하드웨어로서 구현되는지 또는 소프트웨어로서 구현되는지는 전체 시스템에 부과된 설계 제약들 및 특정한 애플리케이션에 의존한다. 당업자는 설명된 기능을 각각의 특정 애플리케이션에 대해 다양한 방식들로 구현할 수도 있지만, 그러한 구현 결정들이 본 출원의 범위로부터의 일탈을 야기하는 것으로서 해석되지는 않아야 한다.

[0164] 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 다중의 이용들을 갖는 집적 회로 디바이스들과 같은 임의의 다양한 디바이스들에서 구현될 수도 있다. 모듈들 또는 컴포넌트들로서 설명된 임의의 특징들은 집적된 로직 디바이스에서 함께 또는 별개지만 상호운용가능한 로직 디바이스들로서 별도로 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 패키징 재료들을 포함할 수도 있는 컴퓨터 프로그램 제품의 일부를 형성할 수도 있다. 컴퓨터 판독가능 매체는 메모리 또는 데이터 저장 매체, 이를테면 RAM (random access memory) 이를테면, SDRAM (synchronous dynamic random access memory), ROM (read-only memory), NVRAM (non-volatile random access memory), EEPROM (electrically erasable programmable read-only memory), FLASH 메모리, 자기 또는 광학 데이터 저장 매체 등을 포함할 수도 있다. 그 기법들은, 추가적으로 또는 대안적으로, 전파된 신호들 또는 파들과 같이, 명령들 또는 데이터 구조들의 형태로 프로그램 코드를 운반 또는 통신하고 그리고 컴퓨터에 의해 액세스, 판독, 및/또는 실행될 수 있는 컴퓨터 판독가능 통신 매체에 의해 적어도 부분적으로 실현될 수도 있다.

[0165] 프로그램 코드는, 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적 회로들 (ASIC들), 필드 프로그래밍가능 로직 어레이들 (FPGA들), 또는 다른 등가의 집적된 또는 별개의 로직 회로부와 같은 하나 이상의 프로세서들을 포함할 수도 있는 프로세서에 의해 실행될 수도 있다. 그러한 프로세서는 본 개시에서 설명된 기법들 중 임의의 기법을 수행하도록 구성될 수도 있다. 범용 프로세서가 마이크로프로세서일 수도 있지만, 대체예에서, 그 프로세서는 기존의 임의의 프로세서, 제어기, 마이크로제어기, 또는 상태 머신일 수도 있다. 프로세서는 또한 컴퓨팅 디바이스들의 조합, 예를 들면, DSP와 마이크로프로세서의 조합, 복수의 마이크로프로세서들의 조합, DSP 코어와 연계한 하나 이상의 마이크로프로세서들의 조합, 또는 임의의 다른 그러한 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.

[0166] 본 개시의 예시적인 예들은 다음을 포함한다:

[0167] 양태 1:

[0168] 비디오 데이터를 프로세싱하는 방법으로서, 기계 학습 시스템에 의해, 입력 비디오 데이터를 획득하는 단계 - 입력 비디오 데이터는 현재 프레임에 대한 적어도 하나의 휘도 성분을 포함함 -; 및 기계 학습 시스템에 의해, 현재 프레임에 대한 적어도 하나의 휘도 성분을 사용하여 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보 및 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하는 단계를 포함하는, 방법.

[0169] 양태 2:

[0170] 양태 1 에 있어서, 기계 학습 시스템에 의해, 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보 및 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 사용하여, 현재 프레임의 적어도 하나의 휘도 성분에 대한 워핑 파라미터 및 현재 프레임의 하나 이상의 색차 성분들에 대한 하나 이상의 워핑 파라미터들을 결정하

는 단계; 및 현재 프레임의 적어도 하나의 휘도 성분에 대한 워핑 파라미터 및 현재 프레임의 하나 이상의 색차 성분들에 대한 하나 이상의 워핑 파라미터들을 사용하여 현재 프레임에 대한 하나 이상의 인터-프레임 예측들을 결정하는 단계를 더 포함하는, 방법.

[0171] 양태 3:

[0172] 양태 2 에 있어서, 상기 하나 이상의 인터-프레임 예측들은 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 상기 워핑 파라미터 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 하나 이상의 워핑 파라미터들을 사용하여 보간 연산 (interpolation operation) 을 적용함으로써 적어도 부분적으로 결정되는, 방법.

[0173] 양태 4:

[0174] 양태 3 에 있어서, 상기 보간 연산은 삼선형 보간 (trilinear interpolation) 연산을 포함하는, 방법.

[0175] 양태 5:

[0176] 양태 2 내지 4 중 어느 것에 있어서, 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 상기 워핑 파라미터 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 하나 이상의 워핑 파라미터들은 공간-스케일 플로우 (SSF) 워핑 파라미터들을 포함하는, 방법.

[0177] 양태 6:

[0178] 양태 5 에 있어서, 상기 SSF 워핑 파라미터들은 학습된 스케일-플로우 벡터들을 포함하는, 방법.

[0179] 양태 7:

[0180] 양태 1 내지 6 중 어느 것에 있어서, 현재 프레임에 대한 적어도 하나의 휘도 성분을 이용하여 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보 및 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하는 단계는, 현재 프레임의 적어도 하나의 휘도 성분 및 이전 프레임의 적어도 하나의 재구성된 루마 성분에 기초하여, 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보를 결정하는 것; 및 현재 프레임의 적어도 하나의 휘도 성분에 대해 결정된 모션 정보를 이용하여 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하는 것을 포함하는, 방법.

[0181] 양태 8:

[0182] 양태 7 에 있어서, 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 모션 정보는 상기 기계 학습 시스템의 컨볼루션 계층 (convolutional layer) 을 사용하여 결정되는, 방법.

[0183] 양태 9:

[0184] 양태 7 에 있어서, 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 모션 정보는 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대해 결정된 상기 모션 정보를 샘플링함으로써 적어도 부분적으로 결정되는, 방법.

[0185] 양태 10:

[0186] 양태 1 내지 9 중 어느 것에 있어서, 현재 프레임은 비디오 프레임을 포함하는, 방법.

[0187] 양태 11:

[0188] 양태 1 내지 10 중 어느 것에 있어서, 하나 이상의 색차 성분들은 적어도 하나의 색차-청색 성분 및 색차-적색 성분을 포함하는, 방법.

[0189] 양태 12:

[0190] 양태 1 내지 11 중 어느 것에 있어서, 현재 프레임은 휘도-색차 (YUV) 포맷을 갖는, 방법.

[0191] 양태 13:

[0192] 양태 12 에 있어서, YUV 포맷은 YUV 4:2:0 포맷인, 방법.

[0193] 양태 14:

[0194] 비디오 데이터를 프로세싱하기 위한 장치로서, 적어도 하나의 메모리; 및 상기 적어도 하나의 메모리에 커플링

된 하나 이상의 프로세서들을 포함하고, 상기 하나 이상의 프로세서들은, 기계 학습 시스템을 사용하여, 입력 비디오 데이터를 획득하는 것으로서, 상기 입력 비디오 데이터는 현재 프레임에 대한 적어도 하나의 휘도 성분을 포함하는, 상기 입력 비디오 데이터를 획득하는 것을 수행하고; 그리고 상기 기계 학습 시스템을 사용하여, 상기 현재 프레임에 대한 적어도 하나의 휘도 성분을 사용하여 상기 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보 및 상기 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하도록 구성되는, 장치.

[0195] 양태 15:

[0196] 양태 14 에 있어서, 상기 하나 이상의 프로세서들은, 기계 학습 시스템을 사용하여, 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보 및 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 사용하여, 현재 프레임의 적어도 하나의 휘도 성분에 대한 워핑 파라미터 및 현재 프레임의 하나 이상의 색차 성분들에 대한 하나 이상의 워핑 파라미터들을 결정하고; 그리고 현재 프레임의 적어도 하나의 휘도 성분에 대한 워핑 파라미터 및 현재 프레임의 하나 이상의 색차 성분들에 대한 하나 이상의 워핑 파라미터들을 사용하여 현재 프레임에 대한 하나 이상의 인터-프레임 예측들을 결정하도록 구성되는, 장치.

[0197] 양태 16:

[0198] 양태 15 에 있어서, 상기 하나 이상의 인터-프레임 예측들은 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 상기 워핑 파라미터 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 하나 이상의 워핑 파라미터들을 사용하여 보간 연산을 적용함으로써 적어도 부분적으로 결정되는, 장치.

[0199] 양태 17:

[0200] 양태 16 에 있어서, 상기 보간 연산은 삼선형 보간 연산을 포함하는, 장치.

[0201] 양태 18:

[0202] 양태 15 내지 17 중 어느 것에 있어서, 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대한 상기 워핑 파라미터 및 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 하나 이상의 워핑 파라미터들은 공간-스케일 플로우 (SSF) 워핑 파라미터들을 포함하는, 장치.

[0203] 양태 19:

[0204] 양태 18 에 있어서, SSF 워핑 파라미터들은 학습된 스케일-플로우 벡터들을 포함하는, 장치.

[0205] 양태 20:

[0206] 양태 14 내지 19 중 어느 것에 있어서, 현재 프레임에 대한 적어도 하나의 휘도 성분을 이용하여 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보 및 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하기 위해, 상기 하나 이상의 프로세서들은, 현재 프레임의 적어도 하나의 휘도 성분 및 이전 프레임의 적어도 하나의 재구성된 루마 성분에 기초하여, 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보를 결정하고; 그리고 현재 프레임의 적어도 하나의 휘도 성분에 대해 결정된 모션 정보를 이용하여 현재 프레임의 하나 이상의 색차 성분들에 대한 모션 정보를 결정하도록 구성되는, 장치.

[0207] 양태 21:

[0208] 양태 20 에 있어서, 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 모션 정보는 상기 기계 학습 시스템의 컨볼루션 계층을 사용하여 결정되는, 장치.

[0209] 양태 22:

[0210] 양태 20 에 있어서, 상기 현재 프레임의 상기 하나 이상의 색차 성분들에 대한 상기 모션 정보를 결정하기 위해, 상기 하나 이상의 프로세서들은 상기 현재 프레임의 상기 적어도 하나의 휘도 성분에 대해 결정된 상기 모션 정보를 샘플링하도록 구성되는, 장치.

[0211] 양태 23:

[0212] 양태 14 내지 22 중 어느 것에 있어서, 현재 프레임은 비디오 프레임을 포함하는, 장치.

[0213] 양태 24:

[0214] 양태 14 내지 23 중 어느 것에 있어서, 하나 이상의 색차 성분들은 적어도 하나의 색차-청색 성분 및 색차-적색 성분을 포함하는, 장치.

[0215] 양태 25:

[0216] 양태 14 내지 24 에 있어서, 현재 프레임은 휘도-색차 (YUV) 포맷을 갖는, 장치.

[0217] 양태 26:

[0218] 양태 25 에 있어서, YUV 포맷은 YUV 4:2:0 포맷인, 장치.

[0219] 양태 27:

[0220] 양태 14 내지 26 중 어느 것에 있어서, 하나 이상의 프레임을 캡처하도록 구성된 적어도 하나의 카메라를 더 포함하는, 장치.

[0221] 양태 28:

[0222] 양태 14 내지 27 중 어느 것에 있어서, 하나 이상의 프레임을 디스플레이하도록 구성된 적어도 하나의 디스플레이를 더 포함하는, 장치.

[0223] 양태 29:

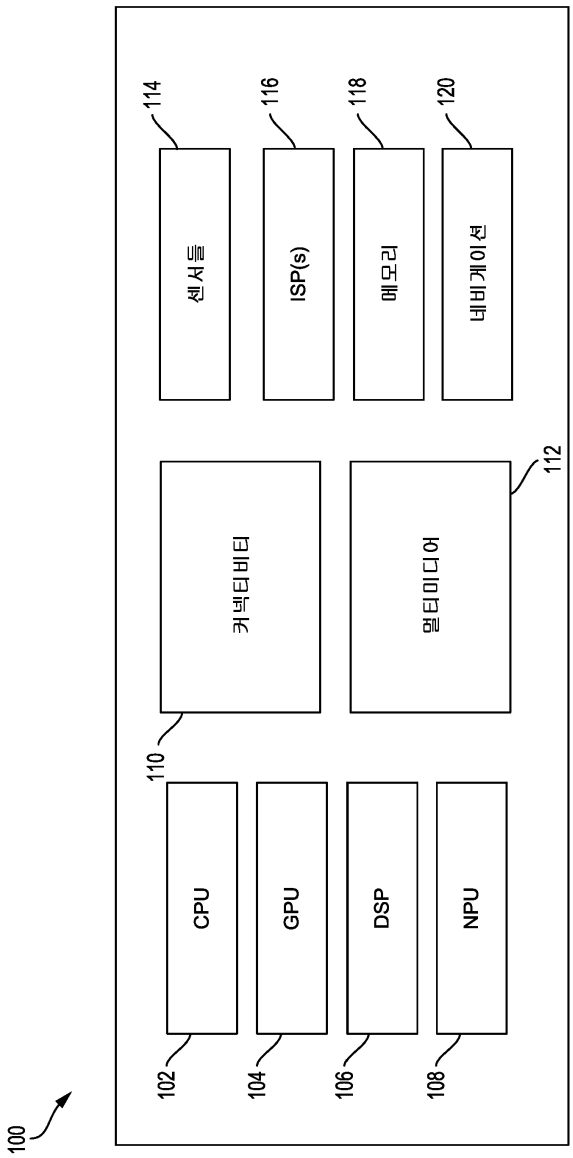
[0224] 양태 14 내지 28 중 어느 것에 있어서, 장치는 모바일 디바이스를 포함하는, 장치.

[0225] 양태 30: 실행될 때, 하나 이상의 프로세서들로 하여금 양태 1 내지 29 의 동작들 중 임의의 것을 수행하게 하는 명령들을 저장하는 컴퓨터 판독가능 저장 매체.

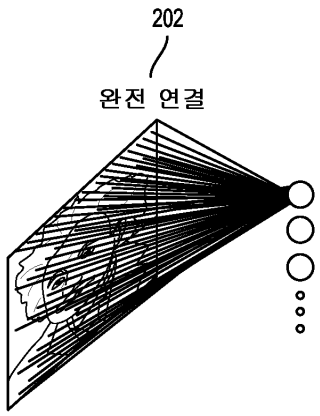
[0226] 양태 31: 양태 1 내지 29 의 동작들 중 임의의 것을 수행하기 위한 수단을 포함하는 장치.

도면

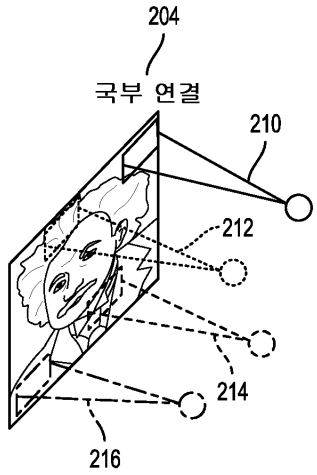
도면1


100
102
CPU
104
GPU
106
DSP
108
NPU
110
112
114
116
118
120
센서들
ISP(s)
메모리
네비게이션
커넥티비티
멀티미디어

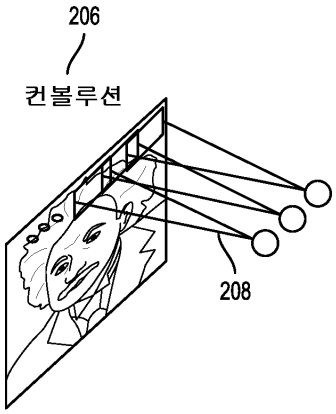
도면2a


202
완전 연결

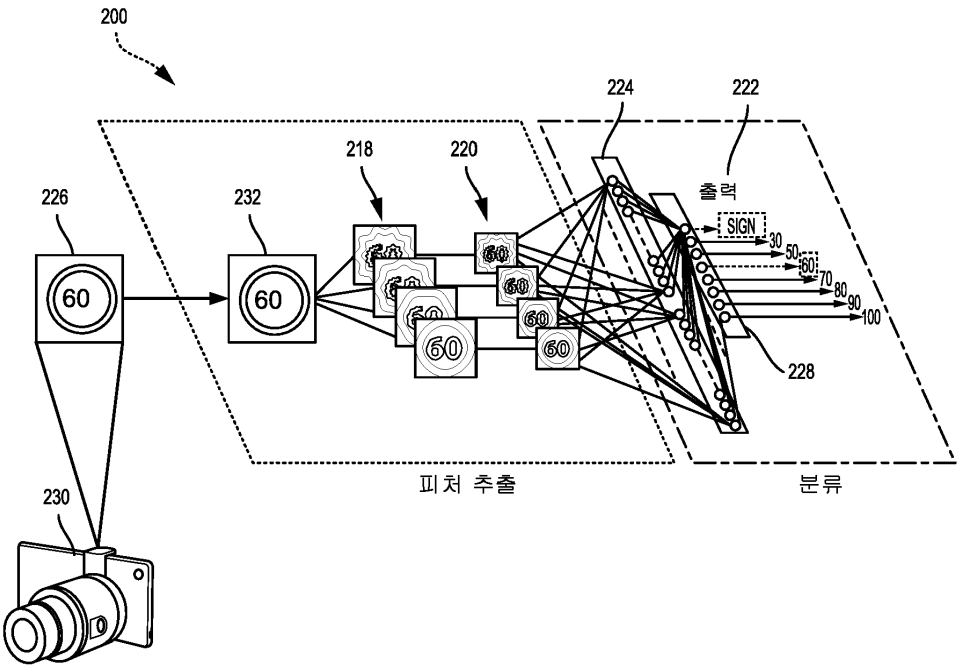
도면2b



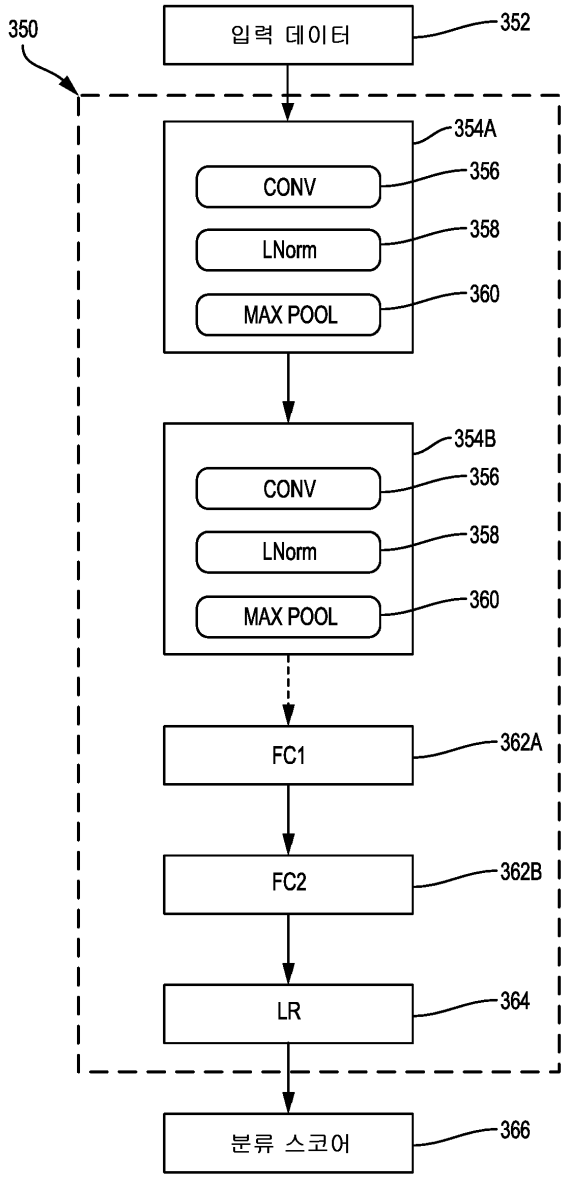
도면2c



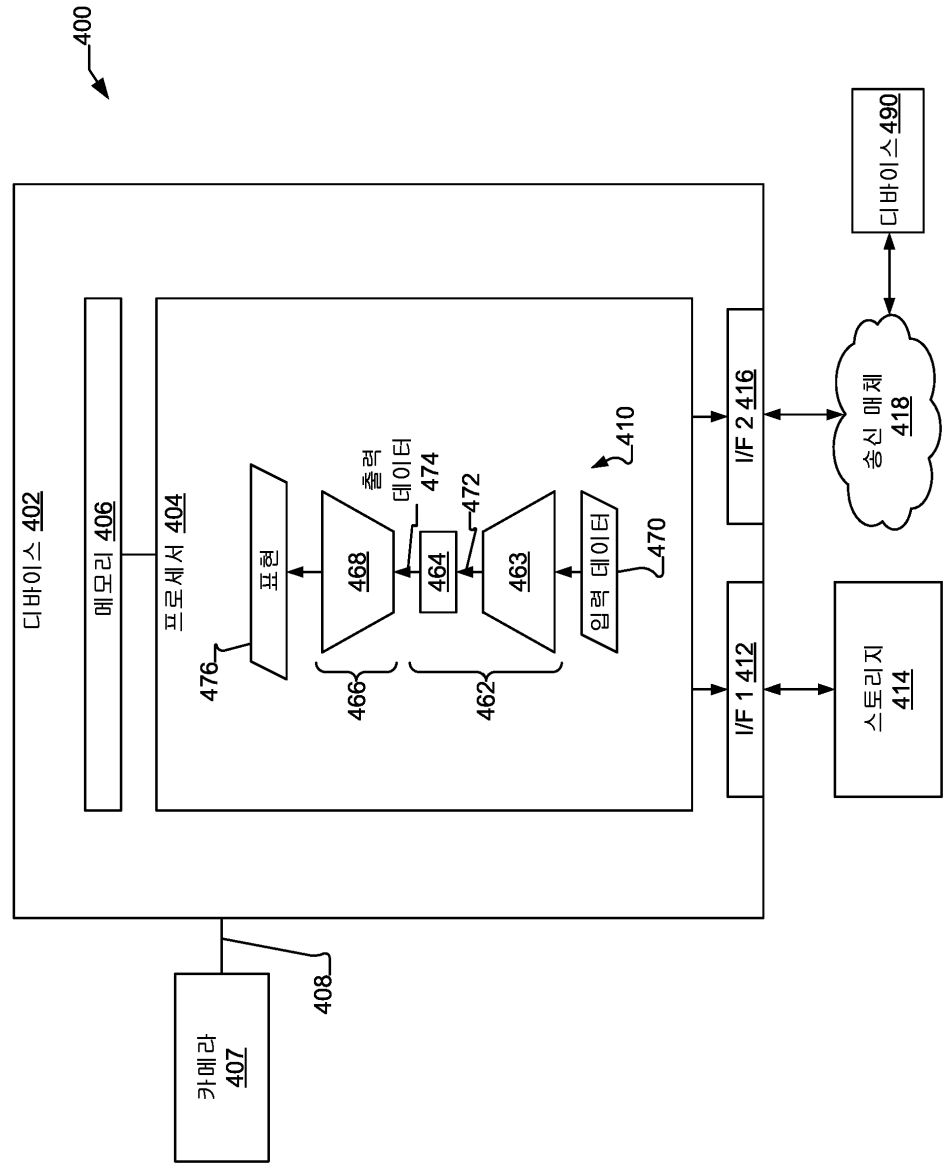
도면2d



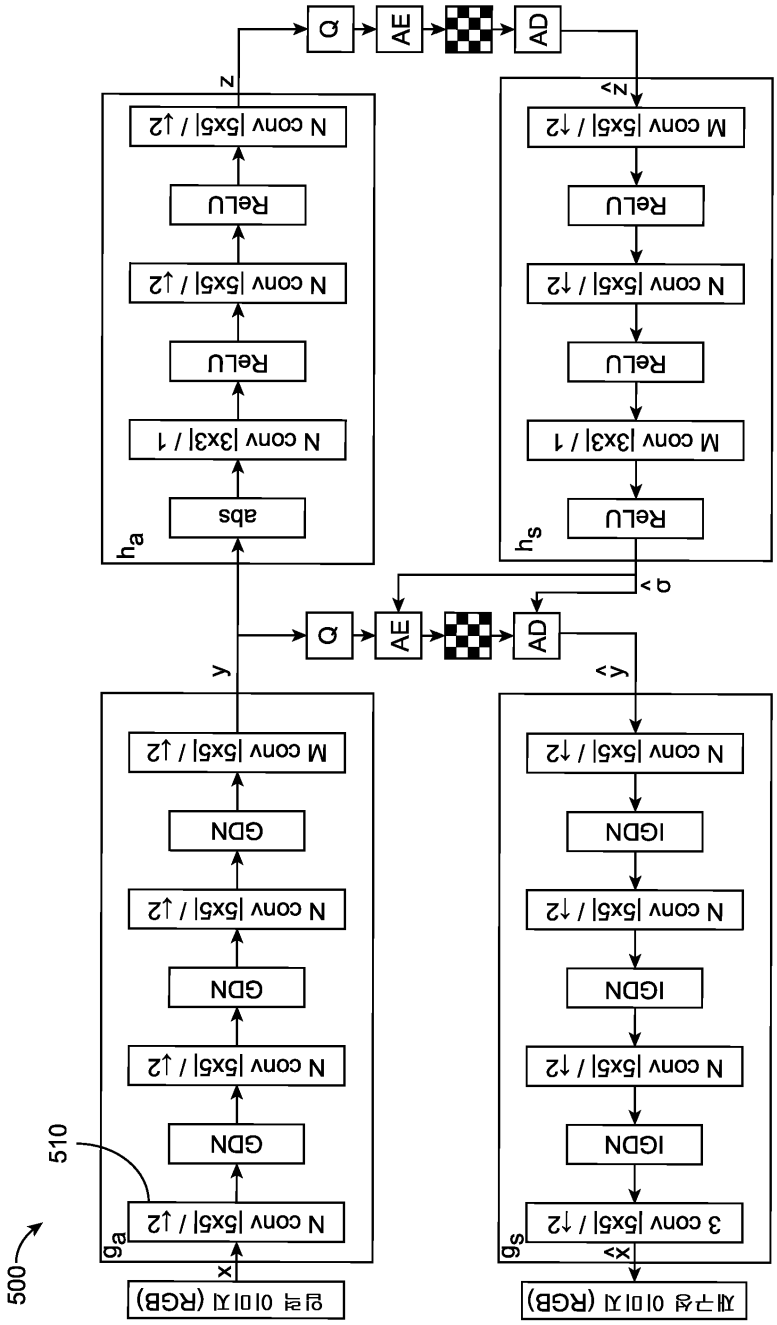
도면3


350
입력 데이터
352
354A
CONV
356
LNorm
358
MAX POOL
360
354B
CONV
356
LNorm
358
MAX POOL
360
FC1
362A
FC2
362B
LR
364
분류 스코어
366

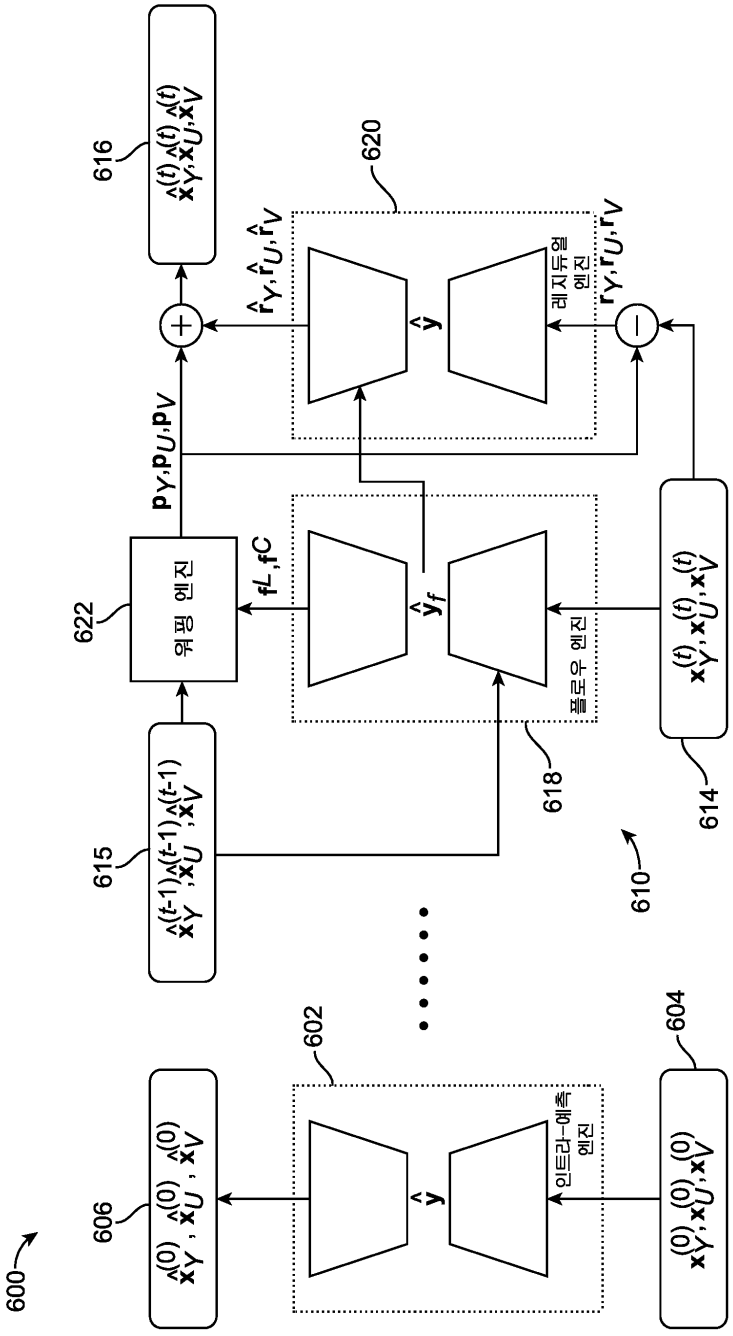
도면4


400
디바이스 402
메모리 406
프로세서 404
표현
468
출력 데이터
474
464
472
463
입력 데이터
410
470
476
466
462
I/F 2 416
I/F 1 412
송신 매체 418
스토리지 414
디바이스 490
카메라 407
408

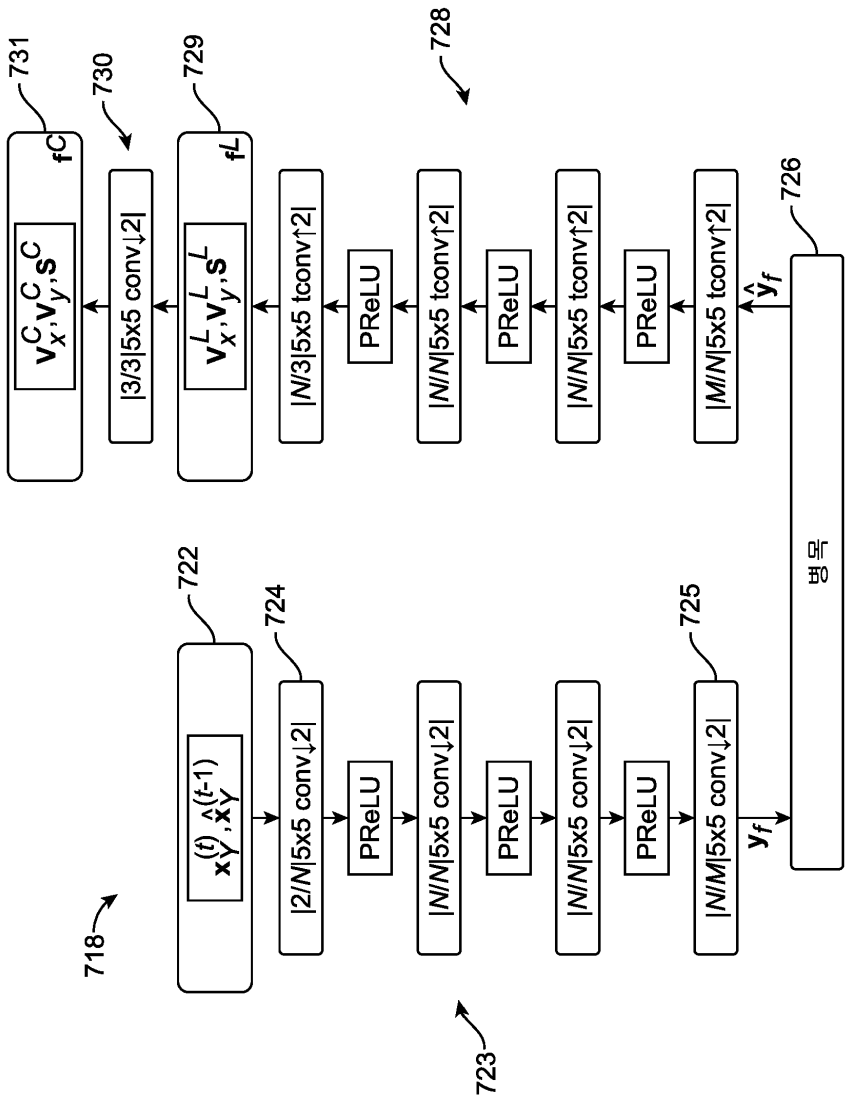
도면5


500
510
입력 이미지 (RGB)
x
N conv |5x5| / ↓2
GDN
N conv |5x5| / ↓2
GDN
N conv |5x5| / ↓2
GDN
M conv |5x5| / ↓2
y
Q
AE
AD
ŷ
N conv |5x5| / ↑2
IGDN
N conv |5x5| / ↑2
IGDN
N conv |5x5| / ↑2
IGDN
3 conv |5x5| / ↑2
x̂
재구성 이미지 (RGB)
g_a
g_s
z
N conv |5x5| / ↓2
ReLU
N conv |5x5| / ↓2
ReLU
N conv |3x3| / 1
abs
h_a
ẑ
M conv |5x5| / ↑2
ReLU
N conv |5x5| / ↑2
ReLU
M conv |3x3| / 1
ReLU
h_s
q
q̂

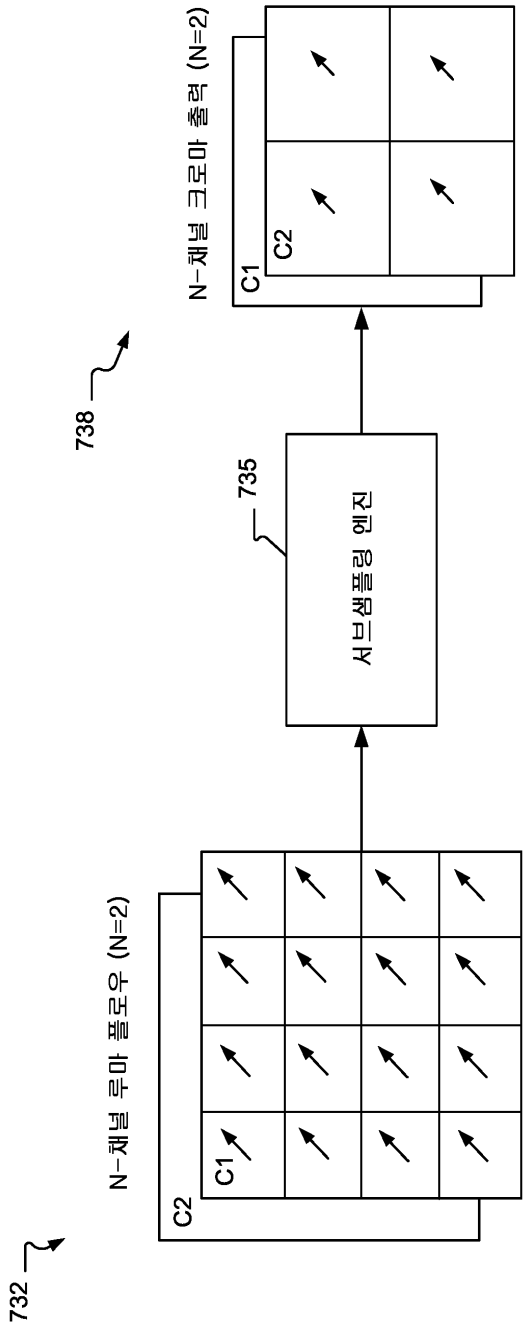
도면6


600
606
602
604
615
618
614
610
622
620
616
인트라-예측 엔진
플로우 엔진
워핑 엔진
레지듀얼 엔진

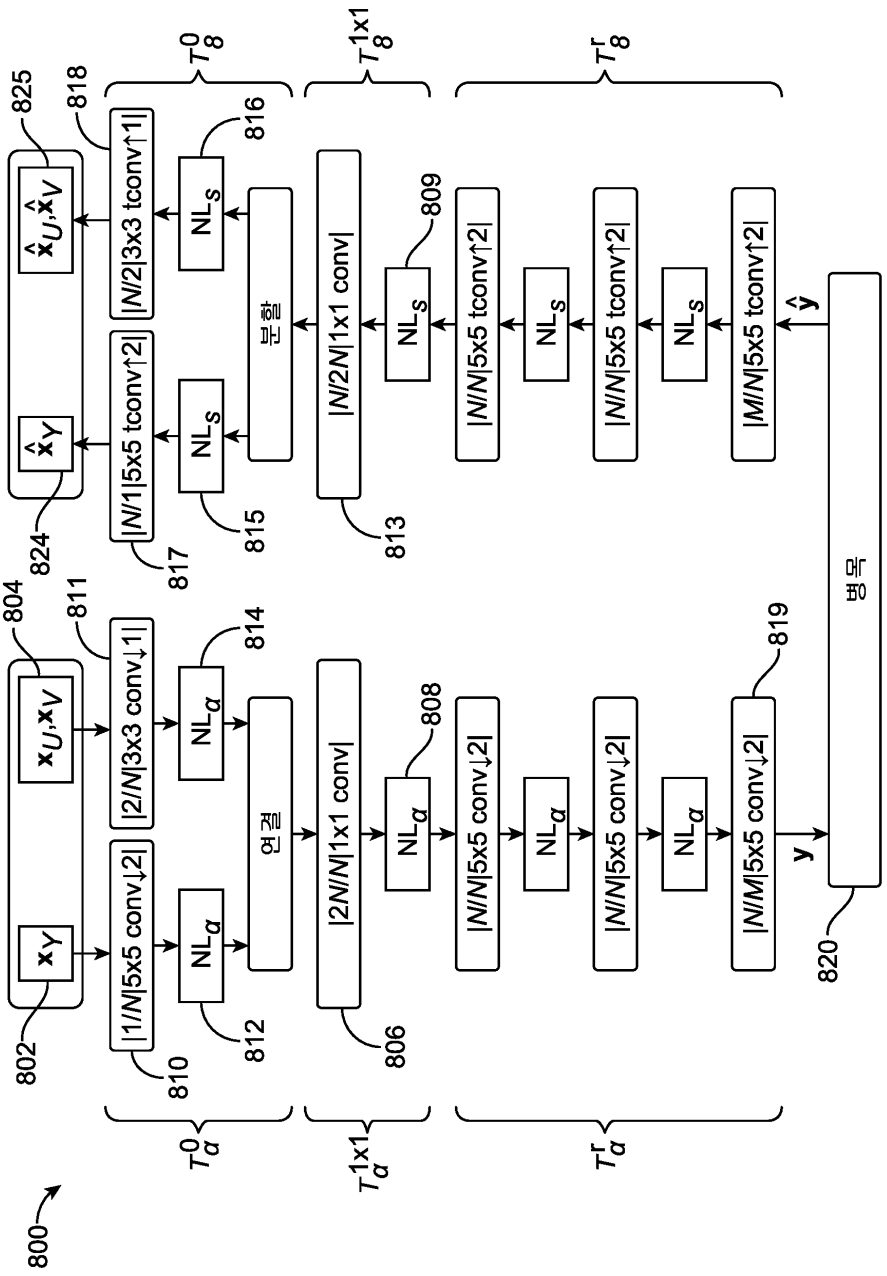
도면7a



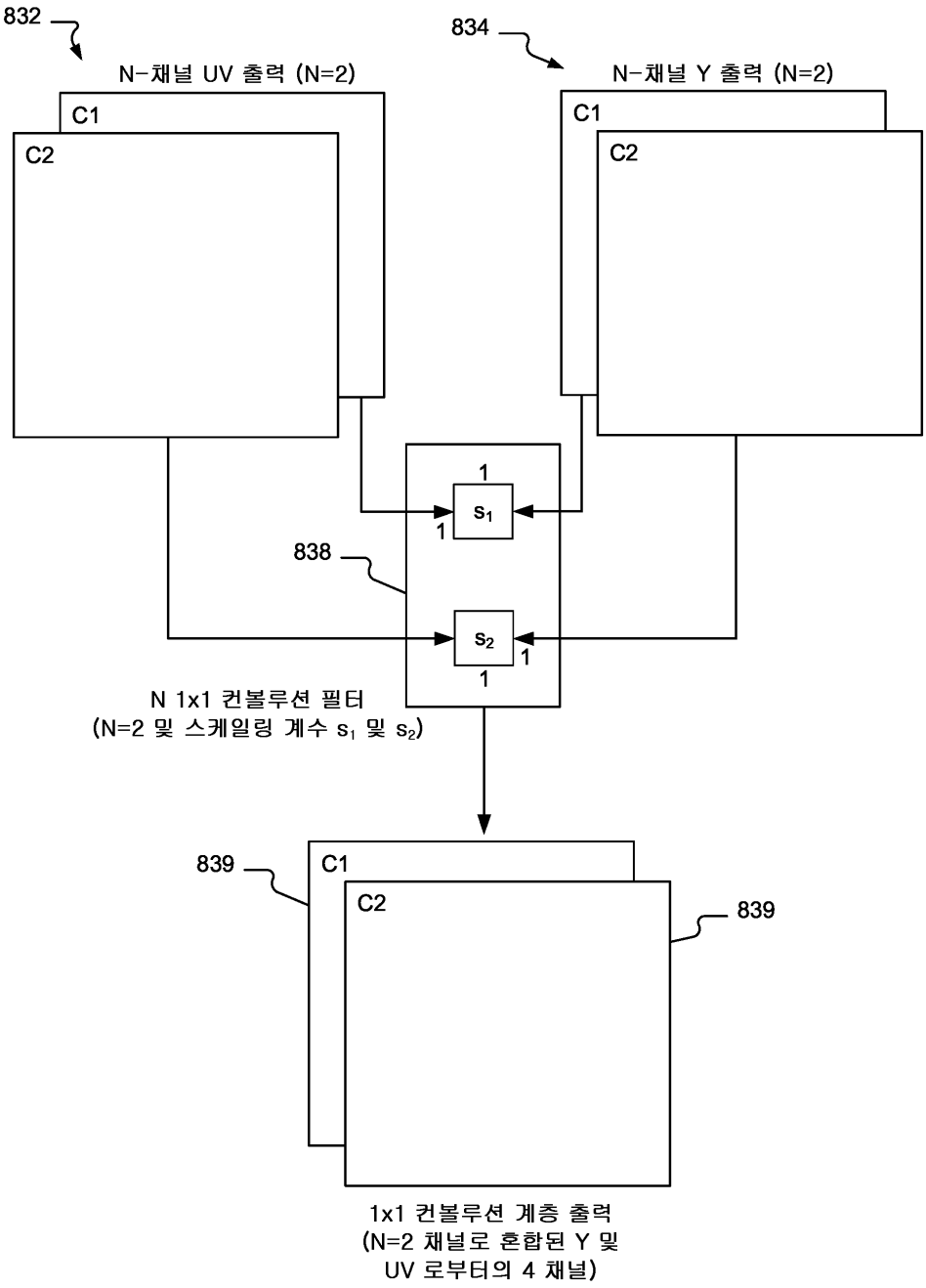
도면7b



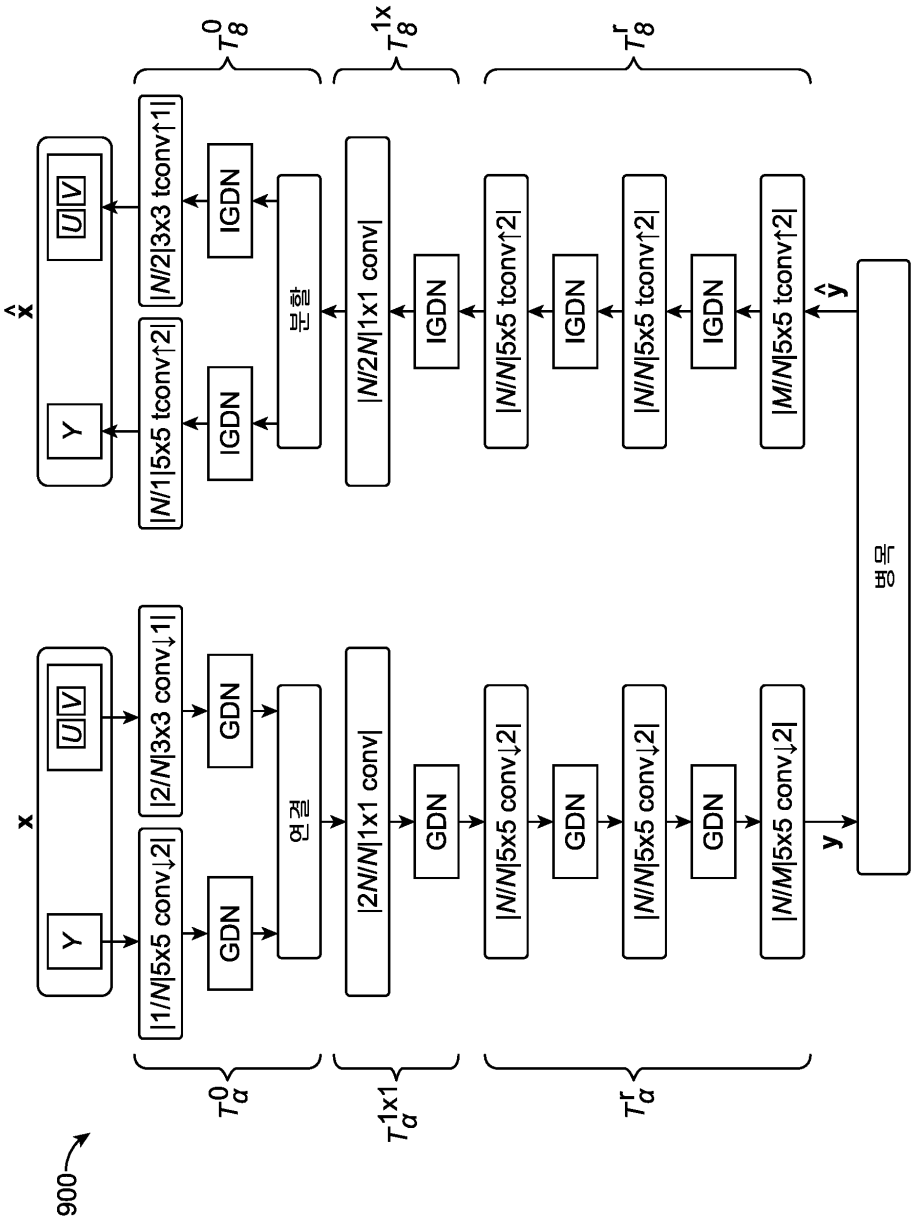
도면8a


800
802
804
806
808
809
810
811
812
813
814
815
816
817
818
819
820
824
825
x_Y
x_U, x_V
$|1/N|5x5\ conv\downarrow 2|$
$|2/N|3x3\ conv\downarrow 1|$
NL_α
연결
$|2N/N|1x1\ conv|$
NL_α
$|N/N|5x5\ conv\downarrow 2|$
NL_α
$|N/N|5x5\ conv\downarrow 2|$
NL_α
$|N/M|5x5\ conv\downarrow 2|$
y
병목
$|M/N|5x5\ tconv\uparrow 2|$
NL_s
$|N/N|5x5\ tconv\uparrow 2|$
NL_s
$|N/N|5x5\ tconv\uparrow 2|$
NL_s
$|N/2N|1x1\ conv|$
NL_s
분할
$|N/1|5x5\ tconv\uparrow 2|$
NL_s
$|N/2|3x3\ tconv\uparrow 1|$
$\hat{x}_Y$
$\hat{x}_U, \hat{x}_V$
$\hat{y}$
T_α^0
T_α^{1x1}
T_α^r
T_8^0
T_8^{1x1}
T_8^r

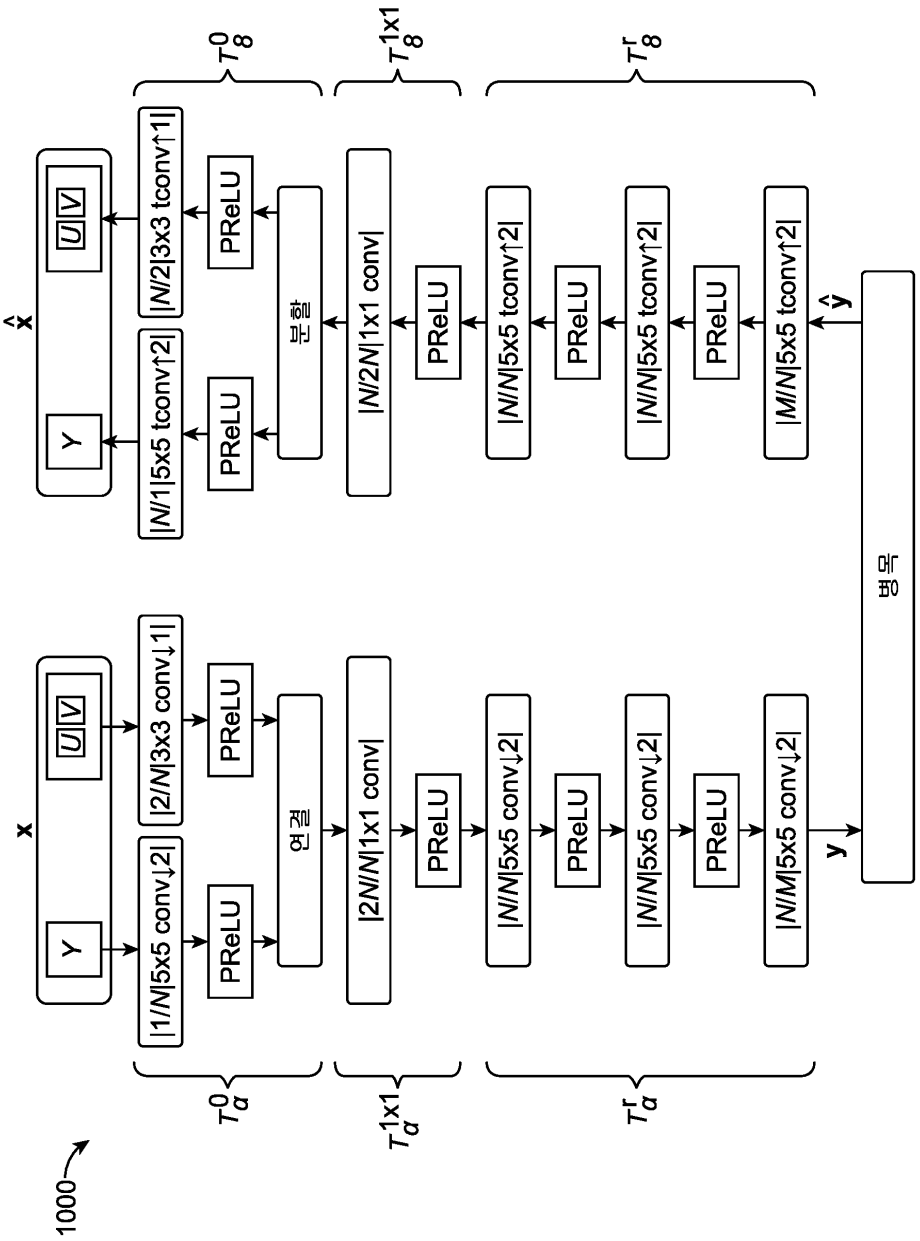
도면8b


832
N-채널 UV 출력 (N=2)
C1
C2
834
N-채널 Y 출력 (N=2)
C1
C2
1
s_1
1
838
s_2
1
1
N 1x1 컨볼루션 필터
(N=2 및 스케일링 계수 s_1 및 s_2)
839
C1
C2
839
1x1 컨볼루션 계층 출력
(N=2 채널로 혼합된 Y 및
UV 로부터의 4 채널)

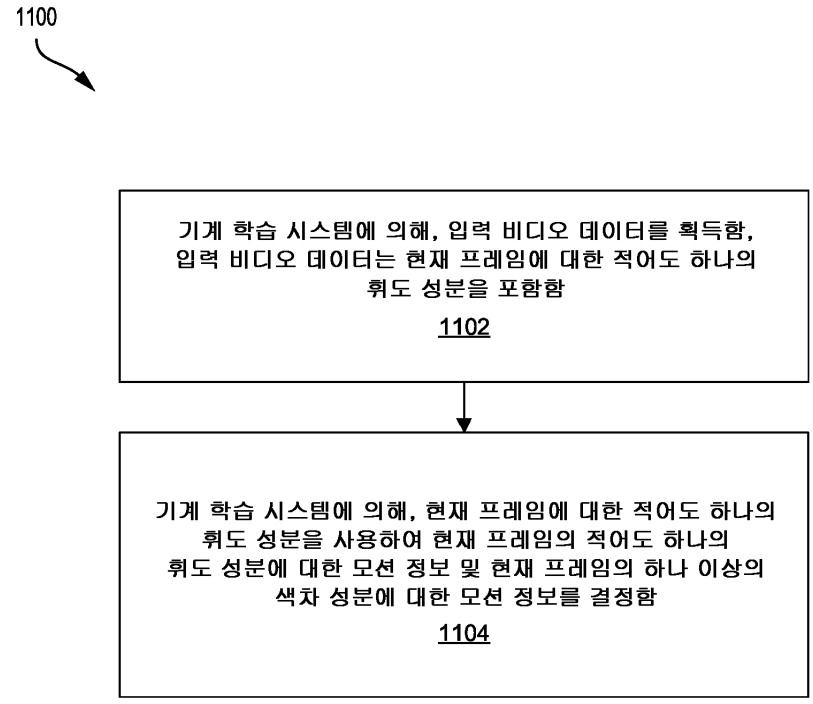
도면9


900
x
Y
U|V
|1/N|5x5 conv↓2|
GDN
|2/N|3x3 conv↓1|
GDN
연결
|2N/N|1x1 conv|
GDN
|N/N|5x5 conv↓2|
GDN
|N/N|5x5 conv↓2|
GDN
|M/N|5x5 conv↓2|
y
양자화
ŷ
|M/N|5x5 tconv↑2|
IGDN
|N/N|5x5 tconv↑2|
IGDN
|N/N|5x5 tconv↑2|
IGDN
|N/2N|1x1 conv|
IGDN
분할
|N/1|5x5 tconv↑2|
IGDN
|N/2|3x3 tconv↑1|
IGDN
Y
U|V
x̂
T_a^0
T_a^{1x1}
T_a^r
T_s^0
T_s^{1x}
T_s^r

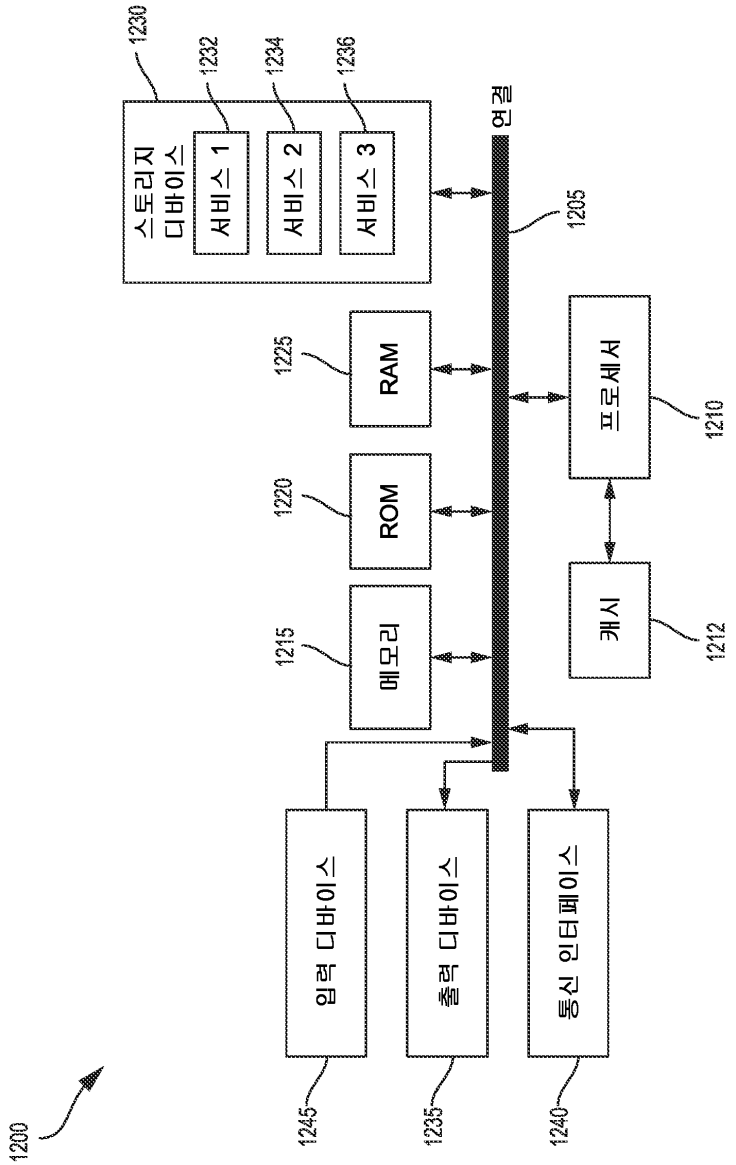
도면10


1000
x
Y
U|V
|1/N|5x5 conv↓2|
PReLU
|2/N|3x3 conv↓1|
PReLU
연결
|2N/N|1x1 conv|
PReLU
|N/N|5x5 conv↓2|
PReLU
|N/N|5x5 conv↓2|
PReLU
|N/M|5x5 conv↓2|
y
병목
ŷ
|N/N|5x5 tconv↑2|
PReLU
|N/N|5x5 tconv↑2|
PReLU
|M/N|5x5 tconv↑2|
|N/2N|1x1 conv|
PReLU
분할
|N/1|5x5 tconv↑2|
PReLU
|N/2|3x3 tconv↑1|
PReLU
x̂
Y
U|V
T_{α}^{0}
T_{α}^{1x1}
T_{α}^{r}
T_{β}^{0}
T_{β}^{1x1}
T_{β}^{r}

도면11


1100
기계 학습 시스템에 의해, 입력 비디오 데이터를 획득함, 입력 비디오 데이터는 현재 프레임에 대한 적어도 하나의 휘도 성분을 포함함
1102
기계 학습 시스템에 의해, 현재 프레임에 대한 적어도 하나의 휘도 성분을 사용하여 현재 프레임의 적어도 하나의 휘도 성분에 대한 모션 정보 및 현재 프레임의 하나 이상의 색차 성분에 대한 모션 정보를 결정함
1104

도면12


1200
1230
스토리지 디바이스
1232
서비스 1
1234
서비스 2
1236
서비스 3
연결
1205
1225
RAM
1220
ROM
1215
메모리
1210
프로세서
1212
캐시
입력 디바이스
출력 디바이스
통신 인터페이스
1245
1235
1240