(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0057844
(43) 공개일자 2020년05월27일

(51) 국제특허분류(Int. Cl.)
H04N 7/01 (2006.01) *G06N 3/04* (2006.01)
G06N 3/08 (2006.01)
(52) CPC특허분류
H04N 7/014 (2013.01)
G06N 3/0454 (2013.01)
(21) 출원번호 10-2018-0140750
(22) 출원일자 2018년11월15일
심사청구일자 2018년11월15일

(71) 출원인
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
강제원
서울특별시 마포구 독막로 145, 104동 904호(창전동,서강쌍용예가아파트)
이정경
서울특별시 양천구 목동동로 350, 506동 1201호(목동, 목동신시가지아파트5단지)
(74) 대리인
특허법인(유한)아이시스

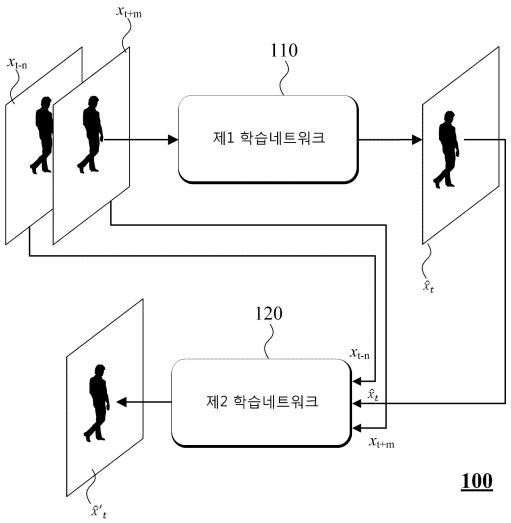
전체 청구항 수 : 총 11 항

(54) 발명의 명칭 **학습네트워크 기반의 비디오 보간 방법 및 비디오 보외 방법**

(57) 요 약

학습네트워크 기반의 비디오 보간 방법은 컴퓨터 장치가 서로 다른 두 개의 영상 프레임을 입력받는 단계, 상기 컴퓨터 장치가 상기 두 개의 영상 프레임을 입력받는 제1 학습네트워크를 이용하여 상기 두 개의 영상 프레임 사이의 특정 시간의 참조 프레임을 생성하는 단계 및 상기 컴퓨터 장치가 상기 두 개의 영상 프레임 및 상기 참조 프레임을 입력받는 제2 학습네트워크를 이용하여 상기 특정 시간의 영상 프레임을 생성하는 단계를 포함한다.

대 표 도 - 도4



(52) CPC특허분류
G06N 3/08 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711076188
부처명 과학기술정보통신부
연구관리전문기관 정보통신기술진흥센터
연구사업명 정보통신·방송 연구개발사업-차세대(UHD) 방송서비스 활성화 기술개발
연구과제명 6DoF지원 초고화질 몰입형 비디오의 압축 및 전송 핵심 기술 개발
기 여 율 1/1
주관기관 이화여자대학교 산학협력단
연구기간 2018.07.01 ~ 2020.12.31

명 세 서

청구범위

청구항 1

컴퓨터 장치가 서로 다른 두 개의 영상 프레임을 입력받는 단계;

상기 컴퓨터 장치가 상기 두 개의 영상 프레임을 입력받는 제1 학습네트워크를 이용하여 상기 두 개의 영상 프레임 사이의 특정 시간의 참조 프레임을 생성하는 단계; 및

상기 컴퓨터 장치가 상기 두 개의 영상 프레임 및 상기 참조 프레임을 입력받는 제2 학습네트워크를 이용하여 상기 특정 시간의 영상 프레임을 생성하는 단계를 포함하는 학습네트워크 기반의 비디오 보간 방법.

청구항 2

제1항에 있어서,

상기 제1 학습네트워크는 시간축에서 일정한 간격을 갖는 두 개의 샘플 영상을 입력받아 상기 두 개의 샘플 영상의 중간 시점에 위치하는 영상을 생성하는 신경망 모델이고,

상기 제2 학습네트워크는 시간축에서 일정한 간격을 갖는 3개의 샘플 영상을 입력받아 상기 3개의 샘플 영상의 중간 시점에 위치하는 영상을 생성하는 신경망 모델인 학습네트워크 기반의 비디오 보간 방법.

청구항 3

제1항에 있어서,

상기 제1 학습네트워크는 GAN(Generative Adversarial Network), ACN(Adaptive convolution network) 또는 컨볼루셔널 인코더-디코더(convolutional encoder-decoder) 중 어느 하나이고, 상기 제2 학습네트워크는 GAN, ACN 또는 컨볼루셔널 인코더-디코더 중 어느 하나인 학습네트워크 기반의 비디오 보간 방법.

청구항 4

제1항에 있어서,

상기 제1 학습네트워크는

상기 두 개의 영상 프레임을 입력받아 제1 특징 벡터를 생성하는 제1 인코더;

상기 두 개의 영상 프레임의 잔차 영상을 입력받아 제2 특징 벡터를 생성하는 제2 인코더;

상기 두 개의 영상 프레임에 대한 상기 제2 특징 벡터를 입력받아 상기 두 개의 영상 프레임 사이 시간에 대한 새로운 특징 벡터를 생성하는 컨볼루션 LSTM(Long Short term memory); 및

상기 제1 특징 벡터와 상기 새로운 특징 벡터를 합산한 값을 입력받아 새로운 영상 프레임을 생성하는 디코더를 포함하는 학습네트워크 기반의 비디오 보간 방법.

청구항 5

제1항에 있어서,

상기 제2 학습네트워크는

상기 두 개의 영상 프레임 및 상기 참조 프레임을 입력받아 제1 특징 벡터를 생성하는 제1 인코더;

상기 두 개의 영상 프레임의 잔차 영상을 입력받아 제2 특징 벡터를 생성하는 제2 인코더;

상기 두 개의 영상 프레임에 대한 상기 제2 특징 벡터를 입력받아 상기 두 개의 영상 프레임 사이 시간에 대한 새로운 특징 벡터를 생성하는 컨볼루션 LSTM(Long Short term memory); 및

상기 제1 특징 벡터와 상기 새로운 특징 벡터를 합산한 값을 입력받아 새로운 영상 프레임을 생성하는 디코더를

포함하는 학습네트워크 기반의 비디오 보간 방법.

청구항 6

컴퓨터에서 제1항 내지 제5항 중 어느 하나의 항에 기재된 학습네트워크 기반의 비디오 보간 방법을 실행하기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록 매체.

청구항 7

컴퓨터 장치가 서로 다른 두 개의 영상 프레임을 입력받는 단계;

상기 컴퓨터 장치가 상기 두 개의 영상 프레임을 입력받는 제1 학습네트워크를 이용하여 상기 두 개의 영상 프레임 이후의 특정 시간의 참조 프레임을 생성하는 단계; 및

상기 컴퓨터 장치가 상기 두 개의 영상 프레임 및 상기 참조 프레임을 입력받는 제2 학습네트워크를 이용하여 상기 특정 시간의 영상 프레임을 생성하는 단계를 포함하는 학습네트워크 기반의 비디오 보외 방법.

청구항 8

제7항에 있어서,

상기 제1 학습네트워크는 시간축에서 일정한 간격을 갖는 두 개의 샘플 영상을 입력받아 상기 두 개의 샘플 영상 이후 시점에 위치하는 영상을 생성하는 신경망 모델이고,

상기 제2 학습네트워크는 시간축에서 일정한 간격을 갖는 3개의 샘플 영상을 입력받아 상기 3개의 샘플 영상 중 마지막 영상의 시점에 위치하는 영상을 생성하는 신경망 모델인 학습네트워크 기반의 비디오 보외 방법.

청구항 9

제7항에 있어서,

상기 제1 학습네트워크는 GAN(Generative Adversarial Network), ACN(Adaptive convolution network) 또는 컨볼루셔널 인코더-디코더(convolutional encoder-decoder) 중 어느 하나이고, 상기 제2 학습네트워크는 GAN, ACN 또는 컨볼루셔널 인코더-디코더 중 어느 하나인 학습네트워크 기반의 비디오 보외 방법.

청구항 10

제7항에 있어서,

상기 제1 학습네트워크는

상기 두 개의 영상 프레임을 입력받아 제1 특징 벡터를 생성하는 제1 인코더;

상기 두 개의 영상 프레임의 잔차 영상을 입력받아 제2 특징 벡터를 생성하는 제2 인코더;

상기 두 개의 영상 프레임에 대한 상기 제2 특징 벡터를 입력받아 상기 두 개의 영상 프레임 이후 시간에 대한 새로운 특징 벡터를 생성하는 컨볼루션 LSTM(Long Short term memory); 및

상기 제1 특징 벡터와 상기 새로운 특징 벡터를 합산한 값을 입력받아 새로운 영상 프레임을 생성하는 디코더를 포함하는 학습네트워크 기반의 비디오 보외 방법.

청구항 11

제7항에 있어서,

상기 제2 학습네트워크는

상기 두 개의 영상 프레임 및 상기 참조 프레임을 입력받아 제1 특징 벡터를 생성하는 제1 인코더;

상기 두 개의 영상 프레임의 잔차 영상을 입력받아 제2 특징 벡터를 생성하는 제2 인코더;

상기 두 개의 영상 프레임에 대한 상기 제2 특징 벡터를 입력받아 상기 두 개의 영상 프레임 이후 시간에 대한 새로운 특징 벡터를 생성하는 컨볼루션 LSTM(Long Short term memory); 및

상기 제1 특징 벡터와 상기 새로운 특징 벡터를 합산한 값을 입력받아 새로운 영상 프레임을 생성하는 디코더를

포함하는 학습네트워크 기반의 비디오 보외 방법.

발명의 설명

기 술 분 야

[0001] 이하 설명하는 기술은 신경망 모델을 사용한 영상 보간법 내지 보외법에 관한 것이다.

배 경 기 술

[0002] 영상 보간법(video interpolation)은 두 개의 영상 프레임을 이용하여 중간 시점의 영상을 예측하는 기법이고, 영상 보외법(video extrapolation)은 연속된 영상 프레임 이후 영상을 예측하는 기법이다.

[0003] 최근 인공신경망과 같은 기계학습 모델을 이용한 다양한 영상 처리 기법이 연구되고 있다. 신경망을 이용한 영상 보간법 등도 연구되고 있다.

선행기술문헌

특허문헌

[0004] (특허문헌 0001) 미국공개특허 US 2018-0137389호

발명의 내용

해결하려는 과제

[0005] 이하 설명하는 기술은 인공신경망에 기반한 영상 보간법 내지 영상 보외법을 제공하고자 한다.

과제의 해결 수단

[0006] 학습네트워크 기반의 비디오 보간 방법은 컴퓨터 장치가 서로 다른 두 개의 영상 프레임을 입력받는 단계, 상기 컴퓨터 장치가 상기 두 개의 영상 프레임을 입력받는 제1 학습네트워크를 이용하여 상기 두 개의 영상 프레임 사이의 특정 시간의 참조 프레임을 생성하는 단계 및 상기 컴퓨터 장치가 상기 두 개의 영상 프레임 및 상기 참조 프레임을 입력받는 제2 학습네트워크를 이용하여 상기 특정 시간의 영상 프레임을 생성하는 단계를 포함한다.

[0007] 학습네트워크 기반의 비디오 보외 방법은 컴퓨터 장치가 서로 다른 두 개의 영상 프레임을 입력받는 단계, 상기 컴퓨터 장치가 상기 두 개의 영상 프레임을 입력받는 제1 학습네트워크를 이용하여 상기 두 개의 영상 프레임 이후의 특정 시간의 참조 프레임을 생성하는 단계 및 상기 컴퓨터 장치가 상기 두 개의 영상 프레임 및 상기 참조 프레임을 입력받는 제2 학습네트워크를 이용하여 상기 특정 시간의 영상 프레임을 생성하는 단계를 포함한다.

발명의 효과

[0008] 이하 설명하는 기술은 계층적 인공신경망을 이용하여 정확한 영상 보간 내지 영상 보외가 가능하다.

도면의 간단한 설명

[0009] 도 1은 컨볼루셔널 계층에 대한 예이다.

도 2는 풀링 계층에 대한 예이다.

도 3은 컨볼루셔널 인코더-디코더에 대한 예이다.

도 4는 인공신경망에 기반한 영상 보간 과정에 대한 예이다.

도 5는 영상 생성 네트워크에 대한 예이다.

도 6은 인공신경망에 기반한 영상 보외 과정에 대한 예이다.

도 7은 영상 생성 네트워크에 대한 예이다.

도 8은 영상 처리 장치에 대한 구성의 예이다.

발명을 실시하기 위한 구체적인 내용

[0010] 이하 설명하는 기술은 다양한 변경을 가할 수 있고 여러 가지 실시례를 가질 수 있는 바, 특정 실시례들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 이하 설명하는 기술을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 이하 설명하는 기술의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0011] 제1, 제2, A, B 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 해당 구성요소들은 상기 용어들에 의해 한정되지는 않으며, 단지 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 이하 설명하는 기술의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

[0012] 본 명세서에서 사용되는 용어에서 단수의 표현은 문맥상 명백하게 다르게 해석되지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함한다" 등의 용어는 설시된 특징, 개수, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 의미하는 것이지, 하나 또는 그 이상의 다른 특징들이나 개수, 단계 동작 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 배제하지 않는 것으로 이해되어야 한다.

[0013] 도면에 대한 상세한 설명을 하기에 앞서, 본 명세서에서의 구성부들에 대한 구분은 각 구성부가 담당하는 주기능 별로 구분한 것에 불과함을 명확히 하고자 한다. 즉, 이하에서 설명할 2개 이상의 구성부가 하나의 구성부로 합쳐지거나 또는 하나의 구성부가 보다 세분화된 기능별로 2개 이상으로 분화되어 구비될 수도 있다. 그리고 이하에서 설명할 구성부 각각은 자신이 담당하는 주기능 이외에도 다른 구성부가 담당하는 기능 중 일부 또는 전부의 기능을 추가적으로 수행할 수도 있으며, 구성부 각각이 담당하는 주기능 중 일부 기능이 다른 구성부에 의해 전담되어 수행될 수도 있음은 물론이다.

[0014] 또, 방법 또는 동작 방법을 수행함에 있어서, 상기 방법을 이루는 각 과정들은 문맥상 명백하게 특정 순서를 기재하지 않은 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 과정들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.

[0016] 이하 설명하는 기술은 기계학습(machine learning)모델을 사용하여 영상을 생성하는 기법이다. 이하 설명하는 기술은 기계학습모델을 사용하여 영상을 생성한다. 기계학습모델은 널리 알려진 바와 같이 다양한 모델이 있다. 설명의 편의를 위하여 이하 설명하는 기술은 인공신경망(artificial neural network)을 사용하여 영상을 생성한다고 가정한다. 인공신경망은 다양한 모델이 있다. 예컨대, 인공신경망은 CNN(Convolutional Neural Network), 오코인토더(auto encoder), 컨볼루셔널 인코더-디코더(convolutional encoder/decoder), RNN(Recurrent Neural Network) 등이 있다. 이하 관련 설명에서 사용가능한 인공신경망 모델을 소개한다. 다만 이하 설명에서 언급되지 않은 다양한 다른 대체 모델을 사용하여 영상을 생성할 수 있다.

[0017] 이하 인공신경망을 이용하여 영상을 생성하는 주체는 영상 처리 장치라고 설명하다. 영상 처리 장치는 일정한 데이터 처리 및 연산이 가능한 컴퓨터 장치에 해당한다. 예컨대, 영상 처리 장치는 PC, 스마트기기, 서버 등과 같은 장치로 구현될 수 있다. 영상 처리 장치는 사전에 학습된 인공신경망 모델을 이용하여 입력 영상을 처리한다. 영상 처리 장치는 입력 영상을 기준으로 보간된(interpolated) 영상 내지 보외된(extrapolated) 영상을 생성할 수 있다.

[0018] 영상 처리 장치는 컨볼루셔널 인코더-디코더를 사용하여 입력 영상에 대한 보간된 영상 내지 보외된 영상을 생성할 수 있다. 기본적으로 사용될 수 있는 컨볼루셔널 인코더-디코더 간략하게 설명한다.

[0019] CNN은 컨볼루셔널(합성곱) 계층 (convolution layer)와 풀링 계층 (pooling layer)이 여러 층으로 이루어진 네트워크이다. 딥러닝 네트워크를 구현하는 모델 중 하나이다. 컨볼루셔널 계층을 통해 입력 이미지에 필터링을 수행하고 그 결과로 특징맵 (feature map)을 추출한다. 특징맵은 다음 층의 입력으로 사용하여 계층 별로 연속하여 처리한다. 학습이 된 네트워크는 처음에는 에지(edge)같은 영상 내 단순한 구성 요소에 반응하고 계측이 깊어질수록 텍스처(texture), 사물 일부(object parts)에 반응하는 특징을 학습하게 된다. 최근 CNN은

AlexNet, VGG16, ResiNet 등과 같이 계층의 깊이를 점점 더해가며 개발이 되고 있다.

[0020] 도 1은 컨볼루셔널 계층에 대한 예이다. 도 1은 컨볼루셔널 계층의 연산을 보여주고 있다. 컨볼루셔널 계층은 입력 이미지에 대한 컨볼루셔널 연산을 통해 특징맵(feature map)을 출력한다. 이때 컨볼루셔널 연산을 수행하는 필터(filter)를 커널(kernel) 이라고도 부른다. 필터의 크기를 필터 크기 또는 커널 크기라고 한다. 커널을 구성하는 연산 파라미터(parameter)를 커널 파라미터(kernel parameter), 필터 파라미터(filter parameter), 또는 가중치(weight)라고 한다. 컨볼루셔널 계층에서는 하나의 입력에 서로 다른 종류의 필터를 사용할 수 있다. 도 1에서 필터의 개수를 n이라고 표시하였다. n은 자연수이다. 이때 하나의 필터가 입력을 처리하는 과정을 컨볼루셔널 채널 (convolution channel)이라고 지칭한다.

[0021] 도 1은 하나의 채널에서 3x3의 커널 크기를 갖는 필터로 컨볼루셔널 연산을 수행하는 과정에 해당한다. 컨볼루셔널 계층은 입력이미지의 특정 영역을 대상으로 컨볼루션 연산을 수행한다. 도 1에서 점선 사각형으로 표시한 영역이 컨볼루션 연산 대상이다. 연산 영역을 윈도우 (window)라고 부른다. 윈도우는 영상의 좌측 상단에서 우측 하단까지 한 칸씩 이동할 수 있고, 한 번에 이동하는 이동 크기를 조절할 수 있다. 이동 크기를 스트라이드(stride)라고 한다. 컨볼루셔널 계층은 입력이미지에서 윈도우를 이동하면서 입력이미지의 모든 영역에 대하여 컨볼루션 연산을 수행한다. 한편 컨볼루셔널 계층은 영상의 가장 자리에 패딩(padding)을 하여 컨볼루션 연산 후 입력 영상의 차원을 유지시킨다.

[0022] 컨볼루셔널 계층에서의 연산 결과로 얻은 특징맵을 서브 샘플링(sub sampling)하는 과정을 풀링(pooling layer)이라고 한다. 도 2는 풀링 계층에 대한 예이다. 도 2에서 풀링 계층은 2×2의 윈도우에서 하나의 샘플을 선택하여 그 출력의 가로 및 세로가 입력의 절반이 되도록 한다. 풀링은 스트라이드와 윈도우의 크기가 갖도록 하는 것일 일반적이다. 도 2는 스트라이드와 윈도우 크기가 모두 2로 설정된 예이다. 풀링 연산은 최대 풀링(max pooling)과 평균 풀링(average pooling) 등이 있다. 최대 풀링은 윈도우 내에서 가장 큰 샘플 값을 선택한다. 평균 풀링은 윈도우에 포함된 값의 평균 값으로 샘플링한다.

[0023] 비선형 연산 계층(nonlinear operation layer)은 뉴런(노드)에서 출력값을 결정하는 계층이다. 비선형 연산 계층은 전달 함수(transfer function)를 사용한다. 전달 함수는 Relu, sigmoid 함수 등이 있다.

[0025] 오토인코더는 인코더(encoder)와 디코더(decoder)를 포함한다. 오토인코더는 비지도 학습(Unsupervised Learning)에 널리 사용된다. 인코더(Encoder)는 입력 데이터(x)가 들어오면 출력으로 잠재 변수 (latent variable)를 제공한다. 잠재 변수는 입력 x의 특징 벡터 (feature vector) 로 사용할 수 있다. 디코더(Decoder)에서는 잠재 변수로부터 형성한 예측 값을 출력으로 한다.

[0026] 도 3은 컨볼루셔널 인코더-디코더에 대한 예이다. 컨볼루셔널 인코더-디코더는 컨볼루셔널 인코더 및 컨볼루셔널 디코더로 구성된다. 컨볼루셔널 인코더-디코더는 컨볼루션널 계층과 역컨볼루션널 계층(deconvolutional layer)의 쌍으로 이루어진 네트워크 구조이다. 도 3은 각각 5개의 컨볼루셔널 계층과 5개의 역컨볼루셔널 계층을 갖는 구조이다. 컨볼루셔널 인코더는 컨볼루셔널 계층과 풀링 계층을 포함한다. 역컨볼루셔널 계층은 역 컨볼루셔널 계층과 언풀링 계층(unpooling layer)을 포함한다.

[0027] 역컨볼루셔널 계층은 컨볼루셔널 계층의 역동작을 수행한다. 역컨볼루셔널 계층은 컨볼루셔널 계층과 반대 방향으로 컨볼루션 연산을 수행한다. 역컨볼루셔널 계층은 입력으로 특징맵을 받아 커널을 이용한 컨볼루션 연산으로 출력 영상을 생성한다. 스트라이드를 1로 하면 역컨볼루셔널 계층은 특징맵의 가로, 세로 크기가 출력의 가로, 세로와 동일한 영상을 출력한다. 스트라이드를 2로 하면 역컨볼루셔널 계층은 특징맵의 가로, 세로 크기 대비 절반 크기의 영상을 출력한다.

[0028] 언풀링 계층은 풀링 계층의 반대 방향으로 업샘플링(upsampling)을 진행한다. 언풀링 계층은 풀링 계층과 다르게 반대로 차원을 확대하는 역할을 한다. 예컨대, 도 3과는 반대로 언풀링 계층은 하나의 샘플을 2×2의 윈도우로 업 샘플링할 수 있다.

[0029] 컨볼루셔널 인코더-디코더는 오토인코더와 유사하게 입력 - 특징벡터 - 출력을 제공한다. 다만 오토인코더와 다르게 컨볼루셔널 인코더-디코더에서 컨볼루셔널 인코더는 입력 영상에 대한 특징맵을 출력하고, 컨볼루셔널 디코더는 입력받는 특징맵에 대한 영상을 출력한다.

[0031] 비디오 보간법과 보외법에 대하여 설명한다. 시간축 위치하는 특정한 영상들을 기준으로 다른 시간에 위치하는

영상을 생성할 수 있다. (1) 비디오 보간법 (Video Interpolation)은 과거 및 미래의 비디오로부터 현재의 비디오를 추정하는 방법이다. 비디오 보간 방법은 일반적으로 시간 t-1과 t+1의 프레임이 주어질 때 시간 t에서의 프레임을 예측하여 생성한다. (2) 비디오 보외법 (Video Extrapolation)은 과거 및 현재의 비디오로부터 미래의 비디오를 예측하는 방법이다. 미래 프레임 예측은 일반적으로 t=0부터 t=n까지의 프레임을 주어질 때 t=n+1부터 t=m까지의 프레임을 생성한다.

[0033] 영상 처리 장치는 두 개의 학습네트워크(제1 학습네트워크 및 제2 학습네트워크)를 이용하여 영상을 처리한다. 제1 학습네트워크는 시간축에서 두 개의 영상 프레임을 입력받아 영상을 생성한다. 제1 학습네트워크는 최초 입력된 두 개의 영상 프레임을 기준으로 참조 프레임을 생성한다. 제2 학습네트워크는 최초 입력된 두 개의 영상 프레임과 함께 참조 프레임을 입력받아 최종 영상 프레임을 생성한다. 영상 처리 장치는 학습네트워크로 생성된 참조 프레임을 추가적으로 사용하여 보다 정확하게 예측된 최종 영상 프레임을 생성한다. 제1 학습네트워크 및 제2 학습네트워크는 입력영상을 기준으로 특정 영상을 생성하는 인공신경망에 해당한다. 제1 학습네트워크와 제2 학습네트워크는 동일한 구조의 신경망 모델일 수도 있고, 서로 다른 신경망 모델일 수도 있다. 두 개의 학습네트워크를 이용한 영상 보간법 및 영상 보외법에 대하여 설명한다.

[0035] 도 4는 인공신경망에 기반한 영상 보간 과정(100)에 대한 예이다. 시간 t에서 가로 길이 w, 영상의 세로 길이 h, 색 차원 채널의 수 c의 비디오 프레임을 $x_t \in R^{w \times h \times c}$라고 정의한다. 이 경우 시간 0에서 t-1의 프레임으로 구성된 비디오는 아래 수학식 1과 같이 정의할 수 있다.

수학식 1

[0037] $$x_{0:t-1} = [x_0, x_1, \ldots, x_{t-1}]$$

[0039] 제1 학습네트워크(110)는 비디오를 구성하는 프레임 중 두 개의 영상 프레임을 입력받는다. 예컨대, 제1 학습네트워크(110)는 t-1의 영상 프레임과 t+1의 영상 프레임을 이용하여 시간 t의 영상 프레임(참조 프레임)을 생성할 수 있다. 도 4는 제1 학습네트워크(110)가 x_{t-n}의 영상 프레임과 x_{t+m}의 영상 프레임을 입력받아 $\hat{x}_t$의 영상 프레임(참조 프레임)을 생성하는 예를 도시한다. $\hat{x}_t$의 영상 프레임은 x_{t-n}의 영상 프레임과 x_{t+m}의 영상 프레임 사이에 위치하는 영상 프레임이다. $\hat{x}_t$의 영상 프레임은 x_{t-n}의 영상 프레임과 x_{t+m}을 기준으로 예측된 보간된 영상에 해당한다.

[0040] 제2 학습네트워크(120)는 3개의 영상 프레임을 입력받는다. 제2 학습네트워크(120)는 제1 학습네트워크(110)에 입력된 영상 프레임과 제1 학습네트워크(110)가 출력한 영상 프레임을 입력받는다. 즉, 제2 학습네트워크(120)는 x_{t-n}의 영상 프레임, x_{t+m}의 영상 프레임 및 $\hat{x}_t$의 영상 프레임을 입력받는다. 구체적인 신경망 구조에 따라 다를 수 있지만, 기본적으로 제2 학습네트워크(120)는 시간 순서에 따라 차례대로 x_{t-n}, $\hat{x}_t$ 및 x_{t+m}를 입력받는다. 제2 학습네트워크(120)는 x_{t-n}의 영상 프레임과 x_{t+m}에 대한 보간된 영상을 생성한다. 이때 제2 학습네트워크(120)는 제1 학습네트워크(110)가 일차적으로 예측한 $\hat{x}_t$ 영상 프레임을 더 사용하여 예측 정확도가 높다. 제2 학습네트워크(120)는 최종적으로 $\hat{x}'_t$ 영상 프레임을 생성한다. $\hat{x}'_t$ 영상 프레임은 x_{t-n}의 영상 프레임과 x_{t+m}의 영상 프레임 사이에 위치하는 예측된 영상 프레임이다.

[0041] 제1 학습네트워크(110) 및 제2 학습네트워크(120)는 서로 다른 신경망 모델로 구현될 수 있다. 나아가 제1 학습네트워크(110) 및 제2 학습네트워크(120)는 처리하는 영상 신호의 크기가 서로 다를 수 있다. 예컨대, 제1 학습네트워크(110)는 입력 신호는 풀링이나 다운 샘플링을 통하여 크기를 줄여 진행을 하고, 출력 신호를 다시 업

샘플링하여 제2 학습네트워크(120)에 전달할 수 있다. 또는 반대로 제1 학습네트워크(110)는 입력 신호를 업 샘플링하여 크기를 크게 만들고, 출력 신호의 크기를 줄여 제2 학습네트워크(120)에 전달할 수도 있다.

[0042] 제1 학습네트워크(110) 및 제2 학습네트워크(120)는 입력 영상의 특징을 기준으로 특정 영상을 생성한다. 두 개의 입력 영상을 이용하여 특정 영상을 생성하는 인공신경망 모델은 다양하다. 몇 가지 예를 설명한다.

[0044] 제1 학습네트워크(110) 및/또는 제2 학습네트워크(120)는 GAN(Generative Adversarial Network)으로 구현될 수 있다. GAN은 영상을 생성하는 생성기(generator)와 생성한 영상을 실 영상과 구분하는 판별기 (discriminator)의 대립 쌍으로 구성된다. 생성기는 입력 정보에 기반하여 영상을 생성하는 모델이다. 판별기는 분류를 담당하는 모델이다. GAN은 데이터를 생성하는 생성기와 만들어진 데이터를 평가하는 판별기가 서로 대립적으로 학습해 가며 성능을 점차 개선하는 개념에서 출발했다. 생성기와 판별기는 각각 다양한 신경망 중 어느 하나를 이용하여 생성될 수 있다.

[0045] 판별기는 생성된 영상이 진짜인지 또는 가짜인지를 분류한다. 생성기는 잠재 코드 z를 입력받아 영상을 생성하되 판별기를 속이기 위한 영상을 생성하도록 학습된다. 생성기는 G(z) 영상을 생성하고, 판별기는 G(z) 영상에 대한 판별 결과 D(G(z))를 생성한다. 생성기는 (1-D(G(z))를 최소화하는 목적함수를 갖는다. 결국 D(G(z))가 1인 경우 최소가 되는데, 생성기는 판별기가 G(z) 영상을 본래 원본 영상이라고 착각할 수 있도록 학습 된다. 기본적인 GAN에서 다양한 파생 모델이 연구되었다. 따라서 다양한 GAN 모델 중 어느 하나를 이용할 수 있다. 예컨대, 기본적인 GAN, DCGAN(Deep Convolutional GAN), cGAN(Conditional GAN), CycleGAN 등과 같은 모델 중 어느 하나를 이용할 수 있다.

[0046] 제1 학습네트워크(110)가 GAN으로 구현된다면, 생성기는 두 개의 영상 프레임을 입력받아 보간된 영상을 예측하여 생성하고, 판별기는 생성된 영상과 두 개의 영상 프레임 사이(중간)에 위치한 영상과 비교한다. 학습된 제1 학습네트워크(110)는 두 개의 영상 프레임에 대한 보간된 영상을 생성하게 된다.

[0047] 제2 학습네트워크(120)가 GAN으로 구현된다면, 생성기는 3개의 영상 프레임을 입력받아 보간된 영상을 예측하여 생성하고, 판별기는 생성된 영상을 두 개의 영상 프레임 사이(중간)에 위치한 영상과 비교한다. 여기서 3개의 영상은 전술한 x_{t-n}, $\hat{x}_t$ 및 x_{t+m} 영상 프레임이다. 따라서 생성기는 3개의 프레임을 입력받아 x_{t-n} 및 x_{t+m} 영상 프레임 사이의 보간된 영상을 생성하는 모델에 해당한다. 판별기는 x_{t-n} 및 x_{t+m} 영상 프레임 사이에 위치하는 영상 프레임과 생성기가 생성한 영상 프레임을 비교한다.

[0048] 제1 학습네트워크(110) 및/또는 제2 학습네트워크(120)는 ACN(Adaptive convolution network)으로 구현될 수 있다. 종래 기법은 영상 보간을 위하여 영상 간 특징을 예측을 하고 얻은 특징점을 이용하여 픽셀 간의 보간을 적용하는 2 단계의 과정을 수행한다. 이에 반하여 ACN은 커널을 CNN을 이용하여 학습하고 종단 간 학습으로 영상 예측과 픽셀 보간을 한 번에 진행을 한다. ACN은 커널 함수 K가 사전에 마련되어야 한다.

[0049] 제1 학습네트워크(110)가 ACN으로 구현된다면, ACN은 두 개의 영상 프레임을 이용하여 두 개의 영상 프레임 사이의 보간된 영상을 생성한다. ACN은 두 개의 영상 프레임을 기준으로 영상 간 특징을 예측하면서, 픽셀 단위 보간을 수행하여 참조 프레임 $\hat{x}_t$을 생성한다. 제2 학습네트워크(110)가 ACN으로 구현된다면, ACN은 3개의 영상 프레임 x_{t-n}, $\hat{x}_t$ 및 x_{t+m}을 이용하여 두 개의 영상 프레임 사이의 보간된 $\hat{x}'_t$ 영상을 생성한다. ACN은 3개의 영상 프레임을 기준으로 영상 간 특징을 예측하면서, 픽셀 단위 보간을 수행하여 참조 프레임을 생성한다.

[0050] 제1 학습네트워크(110) 및/또는 제2 학습네트워크(120)는 컨볼루셔널 인코더-디코더로 구현될 수 있다. 제1 학습네트워크(110)가 컨볼루셔널 인코더-디코더로 구현된다면, 컨볼루셔널 인코더는 두 개의 영상 프레임을 입력받아 보간된 영상에 대한 특징맵을 생성하고, 컨볼루셔널 디코더는 생성된 특징맵을 기준으로 보간된 영상 $\hat{x}_t$을 생성한다. 제2 학습네트워크(120)가 컨볼루셔널 인코더-디코더로 구현된다면, 컨볼루셔널 인코더는 3개의 영상 프레임 x_{t-n}, $\hat{x}_t$ 및 x_{t+m}을 입력받아 보간된 영상에 대한 특징맵을 생성하고, 컨볼루셔널 디코더는 생성된 특징맵을 기준으로 보간된 영상 $\hat{x}'_t$를 생성한다.

[0052] 제1 학습네트워크(110) 및/또는 제2 학습네트워크(120)는 컨볼루셔널 인코더, LSTM(Long Short Term Memory

networks) 및 컨볼루셔널 디코더로 구현될 수도 있다. 도 5는 영상 생성 네트워크에 대한 예이다. 도 5의 영상 생성 네트워크는 입력 영상을 기준으로 보간된 영상을 생성하는 신경망 모델이다. 도 5는 컨볼루셔널 인코더, LSTM 및 컨볼루셔널 디코더로 구현된 영상 생성 네트워크의 예이다.

[0053] 도 5(A)는 제1 학습네트워크(110)를 구현한 예이다. 도 5(A)의 영상 생성 네트워크를 제1 영상 생성 네트워크라 명명한다. 제1 영상 생성 네트워크(200)는 컨볼루셔널 인코더-디코더와 유사한 구조를 갖는다. 제1 영상 생성 네트워크(200)는 인코더(210), 컨볼루션 LSTM(230) 및 디코더(250)를 포함한다. 인코더(210)는 두 개의 서브 인코더(211 및 212)를 포함한다. 두 개의 서브 인코더는 각각 복수의 컨볼루셔널 계층과 풀링 계층을 갖는 CNN 구조이다. 두 개의 서브 인코더를 각각 제1 서브 인코더(211)과 제2 서브 인코더(212)로 명명한다. 두 개의 서브 인코더는 각각 복수의 컨볼루셔널 계층, 풀링 계층 및 비선형 연산 계층으로 구성될 수 있다.

[0054] 제1 서브 인코더(211)는 두 개의 영상 프레임 x_{t-n} 및 x_{t+m}을 입력받아 처리한다. 제1 서브 인코더(211)는 CNN을 통해 입력 영상의 형태(공간적 레이아웃)를 유지하려고 노력한다. 제1 서브 인코더(211)는 입력 영상에서 움직임의 변화가 없는 객체(또는 영역)를 최대한 유지하기 위한 것이다. 제1 서브 인코더(211)는 t 시간 입력 영상에서 배경에 해당하는 영역에 대한 특징 맵 s(t)를 출력한다. 이를 위해 제1 서브 인코더(211)는 입력 영상에서 움직임 없는 객체들을 유지할 수 있도록 사전에 학습되어야 한다.

[0055] 제2 서브 인코더(212)는 x_{t-n}과 x_{t+m}에 대한 잔차 영상(residual video)을 입력받는다. 제2 서브 인코더(212)는 원본 영상에서 움직임이 있는 객체(또는 영역)에 대한 특징 맵 r(t)를 생성한다. 이를 위해 제2 서브 인코더(212)는 입력 영상의 프레임 사이의 차이(잔차 영상)를 이용하여 사전에 학습되어야 한다.

[0056] LSTM(Long Short term memory)에 대하여 먼저 간략하게 설명한다. RNN은 현재 들어온 입력 데이터와 과거에 입력 받았던 데이터를 학습에 동시에 고려한다. RNN은 LSTM으로 구현될 수 있다. LSTM은 여러 게이트(gate)가 붙어있는 셀(cell)로 이루어져 있다. 해당 셀에 연결된 게이트의 값 또는 가중치(weight) 을 확인하여 어떤 값을 저장할지, 언제 정보를 내보내거나 삭제할지를 결정한다. 각 셀 마다 이 가중치 값을 학습하며 학습 성능을 높인다.

[0057] 컨볼루션 LSTM를 이용하면 시계열 정보를 학습하는 RNN의 구조에 LSTM을 활용하여 비디오 생성할 수 있다. 컨볼루션 LSTM은 입력과 은닉벡터 간 연결을 컨볼루션 필터로 대체하여 기존 LSTM 대비 더 적은 양의 파라미터를 학습시킬 수 있고 지역적 특성 잘 반영할 수 있다.

[0058] 컨볼루션 LSTM(230)은 제2 서브 인코더(212)가 출력하는 특징 벡터를 시간 흐름을 고려하여 일정하게 변경한다. 컨볼루션 LSTM(230)은 RNN의 구조에 LSTM을 활용하여 제2 서브 인코더(212)가 출력하는 특징 벡터를 시간의 흐름에 따라 변경하게 된다. 이를 위해 컨볼루션 LSTM(230)은 사전에 연속적인 훈련 프레임에서 이동 객체에 대한 움직임 내지 특징 벡터로 학습되어야 한다. 보다 구체적인 동작은 이하 설명한다.

[0059] 영상 생성 네트워크(200)는 $x_{0:t-1}$ 비디오 프레임(시퀀스)를 사용한다고 가정한다. "0:t-1"은 "시간 0 ~ t-1"을 의미한다. 이때 잔차 영상 $y_{0:t-1}$은 아래 수학식 2와 같이 정의될 수 있다.

수학식 2

[0061]
$$y_{0:t-1} = [y_0, y_1, \cdots, y_{t-1}]$$
$$= [x_0 - x_{-1}, x_1 - x_0, \cdots, x_{t-1} - x_{t-2}]$$

[0063] 제1 서브 인코더(211)는 각 시간에 x_t로부터 특징 벡터 s_t를 출력한다. 제2 서브 인코더(212)는 각 시간에 y_t로부터 특징 벡터 r_t를 출력한다. 제1 서브 인코더(211)는 두 개의 영상 프레임 x_{t-n}과 x_{t+m}에 대한 특징 벡터 s_t를 출력할 수 있다. 또 제2 서브 인코더(212)는 두 개의 영상 프레임의 잔차 영상(x_{t+m}-x_{t-n})으로부터 특징 벡터 r_t를 출력할 수 있다. $s_t, r_t \in \mathbb{R}^{f_h \times f_w \times f_c}$ 이다. f_h는 필터 맵의 높이, f_w는 필터 맵의 너비, f_c는 필터맵이 개수

를 의미한다. 각 컨볼루셔널 계층에서 $f_w = w/2^{\lambda}$이고, $f_h = h/2^{\lambda}$이다. λ는 풀링 계층의 개수이다. 각 컨볼루셔널 계층의 필터 크기는 3×3일 수 있다. 컨볼루셔널 계층에서 필터 맵의 개수는 $64 \times 2^{\lambda}$일 수 있다. 풀링 계층 다음에 Relu와 같은 전달 함수가 위치할 수 있다. 또한 출력되는 특징 벡터는 일정하게 후처리(normalization)될 수도 있다. 제2 서브 인코더(212)도 제1 서브 인코더(211)와 동일한 구조를 가질 수 있다.

[0064] 컨볼루션 LSTM(230)은 제2 서브 인코더(212)가 출력하는 특징 벡터 r_t를 r_{t+1}의 추정값 $\hat{r}_{t+1}$으로 변경한다. r_{t+1}은 영상 프레임 x_{t-n} 및 x_{t+m} 사이(또는 중간)에 있는 프레임의 특징 벡터이다. LSTM 모듈은 메모리 셀 c_t의 정보를 보유한 상태에서 r_t를 수신한다. 그리고 LSTM 모듈은 수신한 정보를 이용하여 예측을 위한 정보를 업데이트한다. LSTM은 $\hat{r}_{t+1}$을 출력하고, 다음 셀 c_{t+1}은 아래의 수학식 3과 같이 정의될 수 있다.

수학식 3

[0066] $$[\hat{r}_{t+1}, c_{t+1}] = \text{ConvLSTM}(\hat{r}_t, c_t)$$

[0067] $\hat{r}_{t+1}$은 시간 t+1의 잔차 영상 프레임에 대응되는 특징 벡터에 해당한다. ConvLSTM은 컨볼루션 LSTM의 동작에 해당하는 함수를 의미한다.

[0069] 디코더(250)는 제1 서브 인코더(211)의 출력 벡터 s_t 및 컨볼루션 LSTM(230)이 출력하는 출력 벡터 $\hat{r}_{t+1}$를 합산(sum)한 값을 입력받는다. 디코더(250)는 s_t+$\hat{r}_{t+1}$ 를 일정한 영상으로 재구성한다. 디코더(250)의 디코딩 함수 DEC는 아래 수학식 4와 같다. 디코더(250)는 $\hat{x}_t$을 출력한다. $\hat{x}_t$는 x_{t-n} 및 x_{t+m} 사이에 위치하는 것으로 예측된 프레임이다. 디코더(250)는 컨볼루셔널 인코더-디코더의 디코더와 같은 구조를 가질 수 있다.

[0071] 도 5(B)는 제2 학습네트워크(120)를 구현한 예이다. 도 5(B)에서 영상 생성 네트워크(300)는 컨볼루셔널 인코더-디코더와 유사한 구조를 갖는다. 도 5(B)의 영상 생성 네트워크를 제2 영상 생성 네트워크라 명명한다. 제2 영상 생성 네트워크(300)는 컨볼루셔널 인코더-디코더와 유사한 구조를 갖는다. 제2 영상 생성 네트워크(300)는 인코더(310), 컨볼루션 LSTM(330) 및 디코더(350)를 포함한다. 인코더(310)는 두 개의 서브 인코더(311 및 312)를 포함한다. 두 개의 서브 인코더는 각각 복수의 컨볼루셔널 계층과 풀링 계층을 갖는 CNN 구조이다. 두 개의 서브 인코더를 각각 제1 서브 인코더(311)과 제2 서브 인코더(312)로 명명한다. 두 개의 서브 인코더는 각각 복수의 컨볼루셔널 계층, 풀링 계층 및 비선형 연산 계층으로 구성될 수 있다.

[0072] 제1 서브 인코더(311)는 3개의 영상 프레임 x_{t-n}, $\hat{x}_t$ 및 x_{t+m}을 입력받아 처리한다. 제1 서브 인코더(311)는 CNN을 통해 입력 영상의 형태(공간적 레이아웃)를 유지하려고 노력한다. 제1 서브 인코더(311)는 입력 영상에서 움직임의 변화가 없는 객체(또는 영역)를 최대한 유지하기 위한 것이다. 제1 서브 인코더(311)는 t 시간 입력 영상에서 배경에 해당하는 영역에 대한 특징 맵 s(t)를 출력한다. 이를 위해 제1 서브 인코더(311)는 입력 영상에서 움직임 없는 객체들을 유지할 수 있도록 사전에 학습되어야 한다.

[0073] 제2 서브 인코더(312)는 x_{t-n}과 x_{t+m}에 대한 잔차 영상(residual video)을 입력받는다. 제2 서브 인코더(312)는 원본 영상에서 움직임이 있는 객체(또는 영역)에 대한 특징 맵 r(t)를 생성한다. 이를 위해 제2 서브 인코더(312)는 입력 영상의 프레임 사이의 차이(잔차 영상)를 이용하여 사전에 학습되어야 한다.

[0074] 컨볼루션 LSTM(330)은 제2 서브 인코더(312)가 출력하는 특징 벡터를 시간 흐름을 고려하여 일정하게 변경한다. 컨볼루션 LSTM(330)은 RNN의 구조에 LSTM을 활용하여 제2 서브 인코더(312)가 출력하는 특징 벡터를 시간의 흐름에 따라 변경하게 된다. 이를 위해 컨볼루션 LSTM(330)은 사전에 연속적인 훈련 프레임에서 이동 객체에 대한

움직임 내지 특징 벡터로 학습되어야 한다.

[0075] 제1 서브 인코더(311)는 각 시간에 x_t로부터 특징 벡터 s_t를 출력한다. 제2 서브 인코더(312)는 각 시간에 y_t로부터 특징 벡터 r_t를 출력한다. 제1 서브 인코더(311)는 3개의 영상 프레임 $x_{t-n,}$ $\hat{x}_t$ 및 x_{t+mm}에 대한 특징 벡터 s_t를 출력할 수 있다. 또 제2 서브 인코더(312)는 두 개의 영상 프레임의 잔차 영상($x_{t+m}-x_{t-n}$)으로부터 특징 벡터 r_t를 출력할 수 있다. $s_t, r_t \in \mathbb{R}^{f_h \times f_w \times f_c}$ 이다. f_h는 필터 맵의 높이, f_w는 필터 맵의 너비, f_c는 필터맵이 개수를 의미한다. 각 컨볼루셔널 계층에서 $f_w = w/2^{\lambda}$이고, $f_h = h/2^{\lambda}$이다. λ는 풀링 계층의 개수이다. 각 컨볼루셔널 계층의 필터 크기는 3×3일 수 있다. 컨볼루셔널 계층에서 필터 맵의 개수는 $64 \times 2^{\lambda}$일 수 있다. 풀링 계층 다음에 Relu와 같은 전달 함수가 위치할 수 있다. 또한 출력되는 특징 벡터는 일정하게 후처리(normalization)될 수도 있다. 제2 서브 인코더(312)도 제1 서브 인코더(311)와 동일한 구조를 가질 수 있다.

[0076] 컨볼루션 LSTM(330)은 제2 서브 인코더(312)가 출력하는 특징 벡터 r_t를 r_{t+1}의 추정값 $\hat{r}_{t+1}$으로 변경한다. r_{t+1}은 영상 프레임 x_{t-n} 및 x_{t+m} 사이(또는 중간)에 있는 프레임의 특징 벡터이다. r_{t+1}은 $\hat{x}_t$와 동일한 시간의 프레임에 대한 특징 벡터이다.

[0077] 디코더(350)는 제1 서브 인코더(311)의 출력 벡터 s_t 및 컨볼루션 LSTM(330)이 출력하는 출력 벡터 $\hat{r}_{t+1}$를 합산(sum)한 값을 입력받는다. 디코더(350)는 s_t+$\hat{r}_{t+1}$ 를 일정한 영상으로 재구성한다. 디코더(350)의 디코딩 함수 DEC는 아래 수학식 4와 같다. 디코더(350)는 $\hat{x}'_t$을 출력한다. $\hat{x}'_t$는 x_{t-n} 및 x_{t+m} 사이에 위치하는 것으로 예측된 프레임이다. 디코더(350)는 컨볼루셔널 인코더-디코더의 디코더와 같은 구조를 가질 수 있다.

[0079] 도 6은 인공신경망에 기반한 영상 보외 과정(400)에 대한 예이다. 제3 학습네트워크(410)는 비디오를 구성하는 프레임 중 두 개의 영상 프레임을 입력받는다. 도 6은 제3 학습네트워크(410)가 x_t 영상 프레임과 x_{t+1} 영상 프레임을 입력받아 $\hat{x}_{t+2}$의 영상 프레임(참조 프레임)을 생성하는 예를 도시한다. $\hat{x}_{t+2}$의 영상 프레임은 x_{t+1} 영상 프레임 다음에 위치하는 영상 프레임이다. $\hat{x}_{t+2}$ 영상 프레임은 x_t 영상 프레임과 x_{t+1}을 기준으로 보외된 영상에 해당한다.

[0080] 제4 학습네트워크(420)는 3개의 영상 프레임을 입력받는다. 제4 학습네트워크(420)는 제3 학습네트워크(410)에 입력된 영상 프레임과 제3 학습네트워크(410)가 출력한 영상 프레임을 입력받는다. 즉, 제4 학습네트워크(420)는 x_t 영상 프레임, x_{t+1}의 영상 프레임 및 $\hat{x}_{t+2}$의 영상 프레임을 입력받는다. 구체적인 신경망 구조에 따라 다를 수 있지만, 기본적으로 제4 학습네트워크(420)는 시간 순서에 따라 차례대로 x_{t-n},x_{t+m} 및 $\hat{x}_{t+2}$를 입력받는다. 제4 학습네트워크(420)는 x_t 영상 프레임과 x_{t+1}을 기준으로 예측되는 보외된 영상을 생성한다. 이때 제4 학습네트워크(420)는 제3 학습네트워크(410)가 일차적으로 예측한 $\hat{x}_{t+2}$ 영상 프레임을 더 사용하여 예측 정확도가 높다. 제4 학습네트워크(420)는 최종적으로 $\hat{x}'_{t+2}$ 영상 프레임을 생성한다. $\hat{x}'_{t+2}$ 영상 프레임은 $\hat{x}_{t+2}$과 같이 x_{t+1} 영상 프레임 다음에 위치하는 영상 프레임이다.

[0081] 제3 학습네트워크(410) 및 제4 학습네트워크(420)는 서로 다른 신경망 모델로 구현될 수 있다. 나아가 제3 학습네트워크(410) 및 제4 학습네트워크(420)는 처리하는 영상 신호의 크기가 서로 다를 수 있다. 예컨대, 제3 학습네트워크(410)는 입력 신호는 풀링이나 다운 샘플링을 통하여 크기를 줄여 진행을 하고, 출력 신호를 다시 업 샘플링하여 제4 학습네트워크(420)에 전달할 수 있다. 또는 반대로 제3 학습네트워크(410)는 입력 신호를 업 샘플링하여 크기를 크게 만들고, 출력 신호의 크기를 줄여 제4 학습네트워크(420)에 전달할 수도 있다.

[0082] 제3 학습네트워크(410) 및 제4 학습네트워크(420)는 입력 영상의 특징을 기준으로 특정 영상을 생성한다. 두 개

의 입력 영상을 이용하여 특정 영상을 생성하는 인공신경망 모델은 다양하다. 예컨대, 제3 학습네트워크(410) 및/또는 제4 학습네트워크(420)는 GAN, ACN 또는 컨볼루셔널 인코더-디코더 중 어느 하나로 구현될 수 있다.

[0084] 도 7은 영상 생성 네트워크에 대한 예이다. 도 7의 영상 생성 네트워크는 입력 영상을 기준으로 보외된 영상을 생성하는 신경망 모델이다. 도 7은 컨볼루셔널 인코더, LSTM 및 컨볼루셔널 디코더로 구현된 영상 생성 네트워크의 예이다. 도 7(A)는 제3 학습네트워크(410)를 구현한 예이다. 도 7(A)에서 영상 생성 네트워크(200)는 컨볼루셔널 인코더-디코더와 유사한 구조를 갖는다. 도 7(A)의 영상 생성 네트워크를 이하 제3 영상 생성 네트워크라고 명명한다. 제3 영상 생성 네트워크(500)는 인코더(510), 컨볼루션 LSTM(530) 및 디코더(550)를 포함한다. 인코더(510)는 두 개의 서브 인코더(511 및 212)를 포함한다. 두 개의 서브 인코더는 각각 복수의 컨볼루셔널 계층과 풀링 계층을 갖는 CNN 구조이다. 두 개의 서브 인코더를 각각 제1 서브 인코더(511)과 제2 서브 인코더(512)로 명명한다. 두 개의 서브 인코더는 각각 복수의 컨볼루셔널 계층, 풀링 계층 및 비선형 연산 계층으로 구성될 수 있다.

[0085] 제1 서브 인코더(511)는 두 개의 영상 프레임 x_t 및 x_{t+1}을 입력받아 처리한다. 제1 서브 인코더(511)는 CNN을 통해 입력 영상의 형태(공간적 레이아웃)를 유지하려고 노력한다. 제1 서브 인코더(511)는 입력 영상에서 움직임의 변화가 없는 객체(또는 영역)를 최대한 유지하기 위한 것이다. 제1 서브 인코더(511)는 t 시간 입력 영상에서 배경에 해당하는 영역에 대한 특징 맵 s(t)를 출력한다. 이를 위해 제1 서브 인코더(511)는 입력 영상에서 움직임 없는 객체들을 유지할 수 있도록 사전에 학습되어야 한다.

[0086] 제2 서브 인코더(512)는 x_{t+1}과 x_t에 대한 잔차 영상(residual video)을 입력받는다. 제2 서브 인코더(512)는 원본 영상에서 움직임이 있는 객체(또는 영역)에 대한 특징 맵 r(t)를 생성한다. 이를 위해 제2 서브 인코더(512)는 입력 영상의 프레임 사이의 차이(잔차 영상)를 이용하여 사전에 학습되어야 한다.

[0087] 컨볼루션 LSTM(530)은 제2 서브 인코더(512)가 출력하는 특징 벡터를 시간 흐름을 고려하여 일정하게 변경한다. 컨볼루션 LSTM(530)은 RNN의 구조에 LSTM을 활용하여 제2 서브 인코더(512)가 출력하는 특징 벡터를 시간의 흐름에 따라 변경하게 된다. 이를 위해 컨볼루션 LSTM(530)은 사전에 연속적인 훈련 프레임에서 이동 객체에 대한 움직임 내지 특징 벡터로 학습되어야 한다.

[0088] 제1 서브 인코더(511)는 각 시간에 x_t로부터 특징 벡터 s_t를 출력한다. 제2 서브 인코더(512)는 각 시간에 잔차 신호 y_t로부터 특징 벡터 r_t를 출력한다. 제1 서브 인코더(511)는 두 개의 영상 프레임 x_t과 x_{t+1}에 대한 특징 벡터 s_t를 출력할 수 있다. 또 제2 서브 인코더(512)는 두 개의 영상 프레임의 잔차 영상($x_{t+1}-x_t$)으로부터 특징 벡터 r_t를 출력할 수 있다. $s_t, r_t \in \mathbb{R}^{f_h \times f_w \times f_c}$ 이다. f_h는 필터 맵의 높이, f_w는 필터 맵의 너비, f_c는 필터맵이 개수를 의미한다. 각 컨볼루셔널 계층에서 $f_w = w/2^{\lambda}$이고, $f_h = h/2^{\lambda}$이다. λ는 풀링 계층의 개수이다. 각 컨볼루셔널 계층의 필터 크기는 3×3일 수 있다. 컨볼루셔널 계층에서 필터 맵의 개수는 $64 \times 2^{\lambda}$일 수 있다. 풀링 계층 다음에 Relu와 같은 전달 함수가 위치할 수 있다. 또한 출력되는 특징 벡터는 일정하게 후처리(normalization)될 수도 있다. 제2 서브 인코더(512)도 제1 서브 인코더(511)와 동일한 구조를 가질 수 있다.

[0089] 컨볼루션 LSTM(530)은 제2 서브 인코더(512)가 출력하는 특징 벡터 r_t를 r_{t+1}의 추정값 $\hat{r}_{t+1}$으로 변경한다. r_{t+1}은 r_t을 다음 프레임에 있는 특징 벡터이다. LSTM 모듈은 메모리 셀 c_t의 정보를 보유한 상태에서 r_t를 수신한다. 그리고 LSTM 모듈은 수신한 정보를 이용하여 예측을 위한 정보를 업데이트한다. $\hat{r}_{t+1}$은 시간 t+1의 잔차 영상 프레임에 대응되는 특징 벡터에 해당한다.

[0090] 디코더(550)는 제1 서브 인코더(511)의 출력 벡터 s_t 및 컨볼루션 LSTM(530)이 출력하는 출력 벡터 $\hat{r}_{t+1}$를 합산(sum)한 값을 입력받는다. 디코더(550)는 $s_t + \hat{r}_{t+1}$ 를 일정한 영상으로 재구성한다. 디코더(550)의 디코딩 함수 DEC는 아래 수학식 4와 같다. 디코더(550)는 $\hat{x}_{t+2}$ 을 출력한다. $\hat{x}_{t+2}$ 는 시간 t+2에서의 추정된 프레임이다. 디코더(550)는 컨볼루셔널 인코더-디코더의 디코더와 같은 구조를 가질 수 있다.

[0091] 도 7(B)는 제4 학습네트워크(420)를 구현한 예이다. 도 7(B)에서 영상 생성 네트워크(600)는 컨볼루셔널 인코더-디코더와 유사한 구조를 갖는다. 도 7(B)의 영상 생성 네트워크를 이하 제4 영상 생성 네트워크라고 명명한다. 제4 영상 생성 네트워크(600)는 인코더(610), 컨볼루션 LSTM(630) 및 디코더(650)를 포함한다. 인코더(610)는 두 개의 서브 인코더(611 및 312)를 포함한다. 두 개의 서브 인코더는 각각 복수의 컨볼루셔널 계층과 풀링 계층을 갖는 CNN 구조이다. 두 개의 서브 인코더를 각각 제1 서브 인코더(611)과 제2 서브 인코더(612)로 명명한다. 두 개의 서브 인코더는 각각 복수의 컨볼루셔널 계층, 풀링 계층 및 비선형 연산 계층으로 구성될 수 있다.

[0092] 제1 서브 인코더(611)는 3개의 영상 프레임 x_t, x_{t+1} 및 $\hat{x}_{t+2}$를 입력받아 처리한다. 제1 서브 인코더(611)는 CNN을 통해 입력 영상의 형태(공간적 레이아웃)를 유지하려고 노력한다. 제1 서브 인코더(611)는 입력 영상에서 움직임의 변화가 없는 객체(또는 영역)를 최대한 유지하기 위한 것이다. 제1 서브 인코더(611)는 t 시간 입력 영상에서 배경에 해당하는 영역에 대한 특징 맵 s(t)를 출력한다. 이를 위해 제1 서브 인코더(611)는 입력 영상에서 움직임 없는 객체들을 유지할 수 있도록 사전에 학습되어야 한다.

[0093] 제2 서브 인코더(612)는 x_{t+1}과 x_t에 대한 잔차 영상을 입력받는다. 제2 서브 인코더(612)는 원본 영상에서 움직임이 있는 객체(또는 영역)에 대한 특징 맵 r(t)를 생성한다. 이를 위해 제2 서브 인코더(612)는 입력 영상의 프레임 사이의 차이(잔차 영상)를 이용하여 사전에 학습되어야 한다.

[0094] 컨볼루션 LSTM(630)은 제2 서브 인코더(612)가 출력하는 특징 벡터를 시간 흐름을 고려하여 일정하게 변경한다. 컨볼루션 LSTM(630)은 RNN의 구조에 LSTM을 활용하여 제2 서브 인코더(612)가 출력하는 특징 벡터를 시간의 흐름에 따라 변경하게 된다. 이를 위해 컨볼루션 LSTM(630)은 사전에 연속적인 훈련 프레임에서 이동 객체에 대한 움직임 내지 특징 벡터로 학습되어야 한다.

[0095] 제1 서브 인코더(611)는 각 시간에 x_t로부터 특징 벡터 s_t를 출력한다. 제2 서브 인코더(612)는 각 시간에 y_t로부터 특징 벡터 r_t를 출력한다. 제1 서브 인코더(611)는 3개의 영상 프레임 x_t, x_{t+1} 및 $\hat{x}_{t+2}$에 대한 특징 벡터 s_t를 출력할 수 있다. 또 제2 서브 인코더(612)는 두 개의 영상 프레임의 잔차 영상($x_{t+1}-x_t$)으로부터 특징 벡터 r_t를 출력할 수 있다. $s_t, r_t \in \mathbb{R}^{f_h \times f_w \times f_c}$ 이다. f_h는 필터 맵의 높이, f_w는 필터 맵의 너비, f_c는 필터맵이 개수를 의미한다. 각 컨볼루셔널 계층에서 $f_w = w/2^{\lambda}$이고, $f_h = h/2^{\lambda}$이다. λ는 풀링 계층의 개수이다. 각 컨볼루셔널 계층의 필터 크기는 3×3일 수 있다. 컨볼루셔널 계층에서 필터 맵의 개수는 $64 \times 2^{\lambda}$일 수 있다. 풀링 계층 다음에 Relu와 같은 전달 함수가 위치할 수 있다. 또한 출력되는 특징 벡터는 일정하게 후처리(normalization)될 수도 있다. 제2 서브 인코더(612)도 제1 서브 인코더(611)와 동일한 구조를 가질 수 있다.

[0096] 컨볼루션 LSTM(630)은 제2 서브 인코더(612)가 출력하는 특징 벡터 r_t를 r_{t+1}의 추정값 $\hat{r}_{t+1}$으로 변경한다. r_{t+1}은 영상 프레임 x_{t+1n} 다음에 위치하는 프레임의 특징 벡터이다. r_{t+1}은 $\hat{x}_{t+2}$와 동일한 시간의 프레임에 대한 특징 벡터이다.

[0097] 디코더(650)는 제1 서브 인코더(611)의 출력 벡터 s_t 및 컨볼루션 LSTM(630)이 출력하는 출력 벡터 $\hat{r}_{t+1}$를 합산(sum)한 값을 입력받는다. 디코더(650)는 $s_t + \hat{r}_{t+1}$ 를 일정한 영상으로 재구성한다. 디코더(650)의 디코딩 함수 DEC는 아래 수학식 4와 같다. 디코더(650)는 $\hat{x}'_{t+2}$ 을 출력한다. $\hat{x}'_{t+2}$ 는 x_{t+1} 다음에 위치하는 것으로 예측된 프레임이다. 디코더(650)는 컨볼루셔널 인코더-디코더의 디코더와 같은 구조를 가질 수 있다.

[0099] 도 8은 영상 처리 장치(700)에 대한 구성의 예이다. 영상 처리 장치(700)는 전술한 영상 보간법 내지 영상 보외법을 실행하는 장치이다. 영상 처리 장치(700)는 물리적으로 다양한 형태로 구현될 수 있다. 예컨대, 영상 처리 장치(700)는 PC와 같은 컴퓨터 장치, 네트워크의 서버, 영상 처리 전용 칩셋 등의 형태를 가질 수 있다. 컴퓨터 장치는 스마트 기기 등과 같은 모바일 기기를 포함할 수 있다.

[0100] 영상 처리 장치(700)는 저장 장치(710), 메모리(720), 연산장치(730), 인터페이스 장치(740) 및 통신 장치(75

0)를 포함한다.

[0101] 저장 장치(710)는 영상 처리를 위한 신경망 모델을 저장한다. 예컨대, 저장 장치(710)는 영상 보간을 위하여 전술한 제1 학습네트워크 및 제2 학습네트워크를 저장할 수 있다. 저장 장치(710)는 영상 보외를 위하여 제3 학습네트워크 및 제4학습네트워크를 저장할 수 있다. 나아가 저장 장치(710)는 영상 처리에 필요한 프로그램 내지 소스 코드 등을 저장할 수 있다. 저장 장치(710)는 입력된 프레임 및 생성된 프레임을 저장할 수 있다.

[0102] 메모리(720)는 영상 처리 장치(700)가 수신한 입력 프레임 및 영상 생성과정에서 생성되는 데이터 및 정보 등을 저장할 수 있다.

[0103] 인터페이스 장치(740)는 외부로부터 일정한 명령 및 데이터를 입력받는 장치이다. 인터페이스 장치(740)는 물리적으로 연결된 입력 장치 또는 외부 저장 장치로부터 투과 현미경 영상을 입력받을 수 있다. 인터페이스 장치(740)는 영상 처리를 위한 각종 신경망 모델을 입력받을 수 있다. 인터페이스 장치(740)는 신경망 모델 생성을 위한 학습데이터, 정보 및 파라미터값을 입력받을 수도 있다.

[0104] 통신 장치(750)는 유선 또는 무선 네트워크를 통해 일정한 정보를 수신하고 전송하는 구성을 의미한다. 통신 장치(750)는 외부 객체로부터 입력 프레임을 수신할 수 있다. 통신 장치(750)는 각종 신경망 모델 및 모델 학습을 위한 데이터도 수신할 수 있다. 통신 장치(750)는 생성한 영상을 외부 객체로 송신할 수 있다.

[0105] 통신 장치(750) 내지 인터페이스 장치(740)는 외부로부터 일정한 데이터 내지 명령을 전달받는 장치이다. 통신 장치(750) 내지 인터페이스 장치(740)를 입력장치라고 명명할 수 있다.

[0106] 연산 장치(730)는 저장장치(710)에 저장된 신경망 모델 내지 프로그램을 이용하여 영상 보간 또는 영상 보외 과정을 처리한다. 연산 장치(730)는 주어진 학습 데이터를 이용하여 영상 처리 과정에 사용되는 신경망 모델을 학습할 수 있다. 연산 장치(730)는 전술한 과정을 통해 구축된 신경망을 이용하여 입력 영상에 대한 보간된 영상 또는 보외된 영상을 생성할 수 있다. 연산 장치(730)는 데이터를 처리하고, 일정한 연산을 처리하는 프로세서, AP, 프로그램이 임베디드된 칩과 같은 장치일 수 있다.

[0108] 또한, 상술한 바와 같은 영상 보간법 내지 영상 보외법은 컴퓨터에서 실행될 수 있는 실행가능한 알고리즘을 포함하는 프로그램(또는 어플리케이션)으로 구현될 수 있다. 상기 프로그램은 비일시적 판독 가능 매체(non-transitory computer readable medium)에 저장되어 제공될 수 있다.

[0109] 비일시적 판독 가능 매체란 레지스터, 캐쉬, 메모리 등과 같이 짧은 순간 동안 데이터를 저장하는 매체가 아니라 반영구적으로 데이터를 저장하며, 기기에 의해 판독(reading)이 가능한 매체를 의미한다. 구체적으로는, 상술한 다양한 어플리케이션 또는 프로그램들은 CD, DVD, 하드 디스크, 블루레이 디스크, USB, 메모리카드, ROM 등과 같은 비일시적 판독 가능 매체에 저장되어 제공될 수 있다.

[0111] 본 실시례 및 본 명세서에 첨부된 도면은 전술한 기술에 포함되는 기술적 사상의 일부를 명확하게 나타내고 있는 것에 불과하며, 전술한 기술의 명세서 및 도면에 포함된 기술적 사상의 범위 내에서 당업자가 용이하게 유추할 수 있는 변형 예와 구체적인 실시례는 모두 전술한 기술의 권리범위에 포함되는 것이 자명하다고 할 것이다.

부호의 설명

[0112] 110 : 제1 학습네트워크

120 : 제2 학습네트워크

200 : 제1 영상 생성 네트워크

210 : 인코더

211 : 제1 인코더

212 : 제2 인코더

230 : 컨볼루션 LSTM

250 : 디코더

300 : 제2 영상 생성 네트워크

310 : 인코더

311 : 제1 인코더

312 : 제2 인코더

330 : 컨볼루션 LSTM

350 : 디코더

410 : 제1 학습네트워크

420 : 제2 학습네트워크

500 : 제3 영상 생성 네트워크

510 : 인코더

511 : 제1 인코더

512 : 제2 인코더

530 : 컨볼루션 LSTM

550 : 디코더

600 : 제4 영상 생성 네트워크

610 : 인코더

611 : 제1 인코더

612 : 제2 인코더

630 : 컨볼루션 LSTM

650 : 디코더

700 : 영상 처리 장치

710 : 저장장치

720 : 메모리

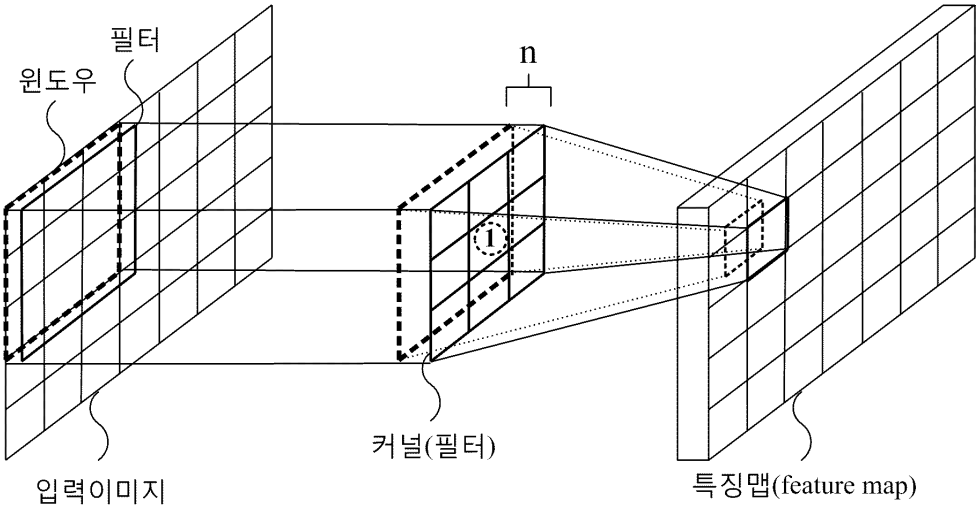
730 : 연산장치

740 : 인터페이스장치

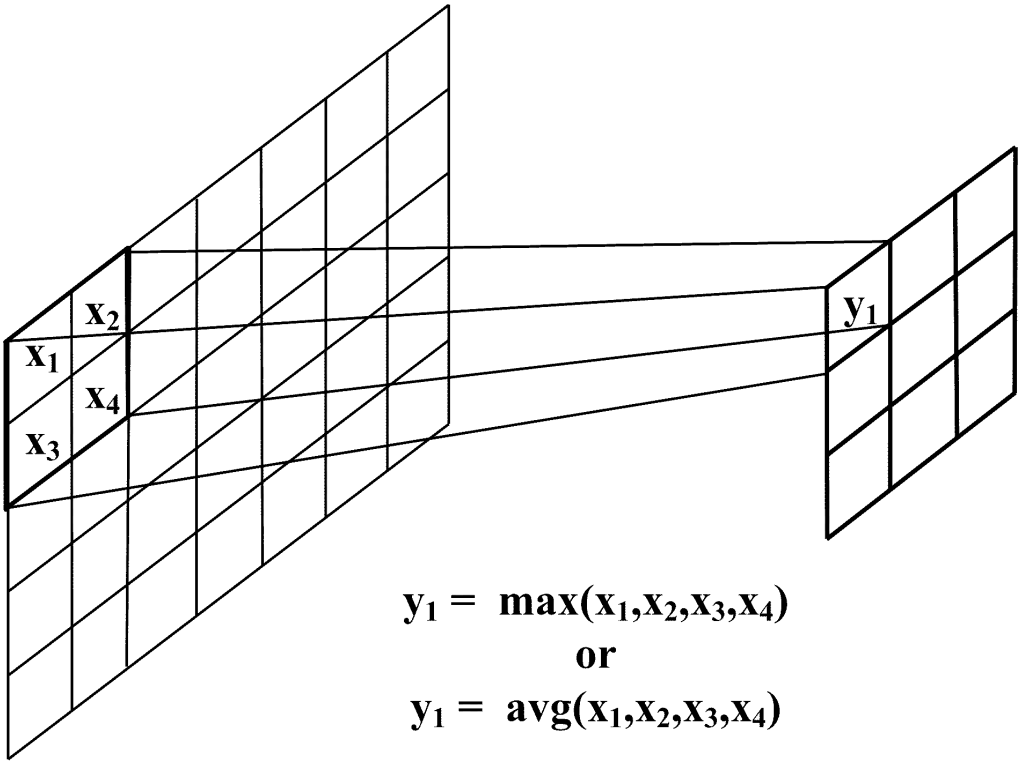
750 : 통신장치

도면

도면1



도면2

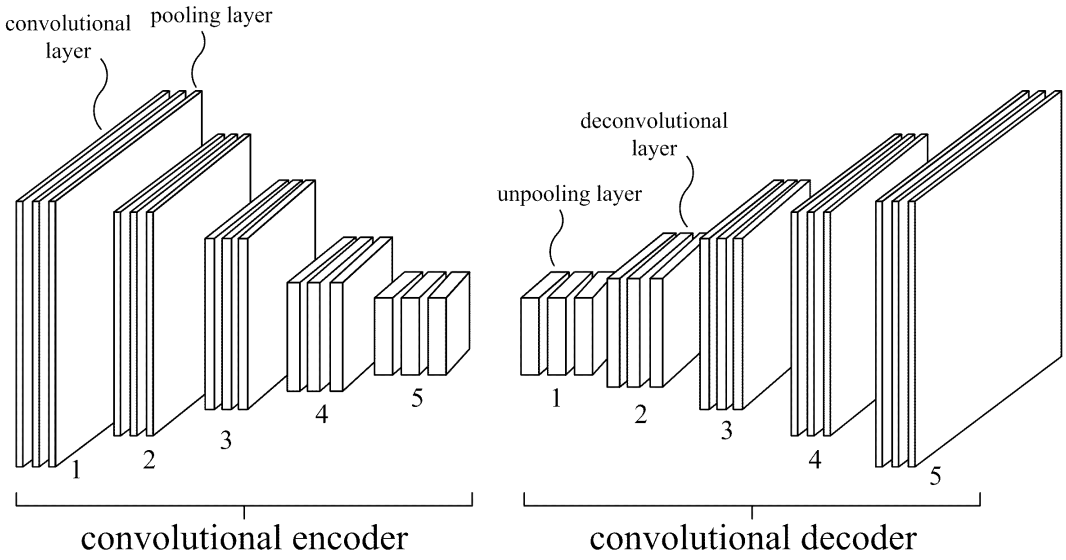


$$y_1 = \max(x_1,x_2,x_3,x_4)$$

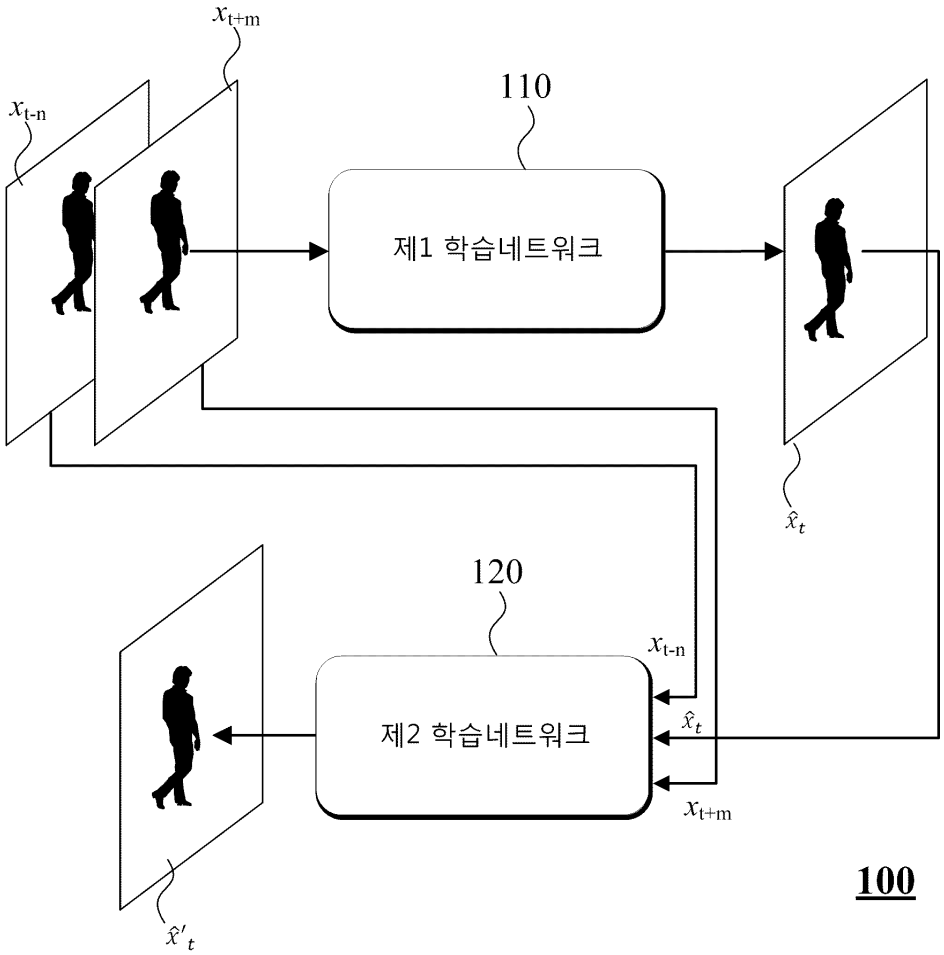
or

$$y_1 = \text{avg}(x_1,x_2,x_3,x_4)$$

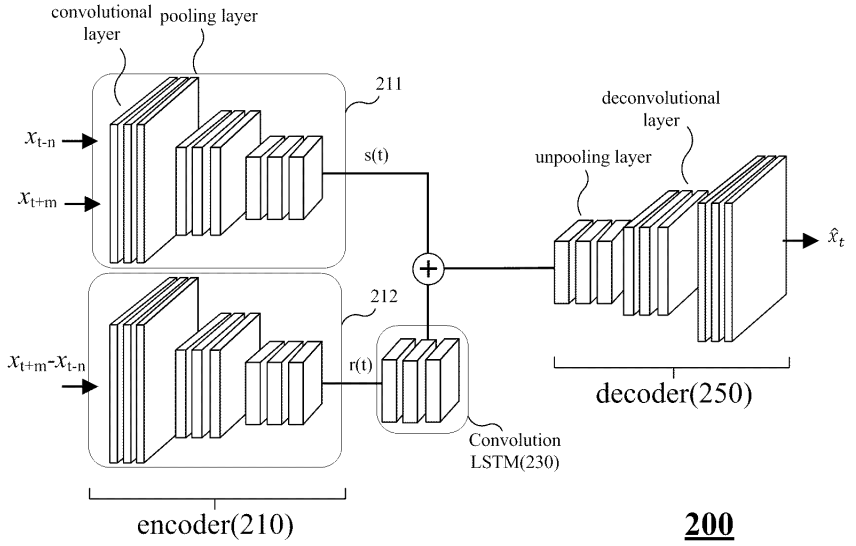
도면3


convolutional layer
pooling layer
deconvolutional layer
unpooling layer
1
2
3
4
5
convolutional encoder
convolutional decoder

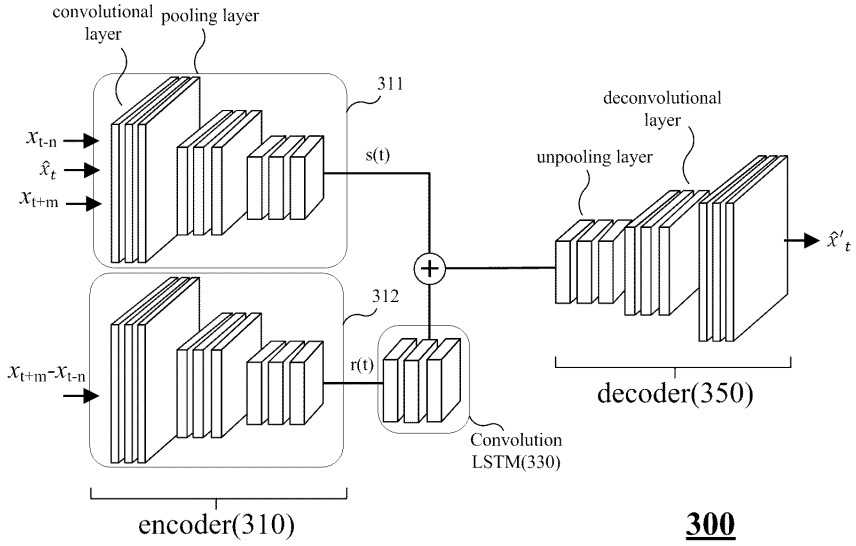
도면4


x_{t+m}
x_{t-n}
110
제1 학습네트워크
$\hat{x}_t$
120
제2 학습네트워크
$\hat{x}'_t$
100

도면5

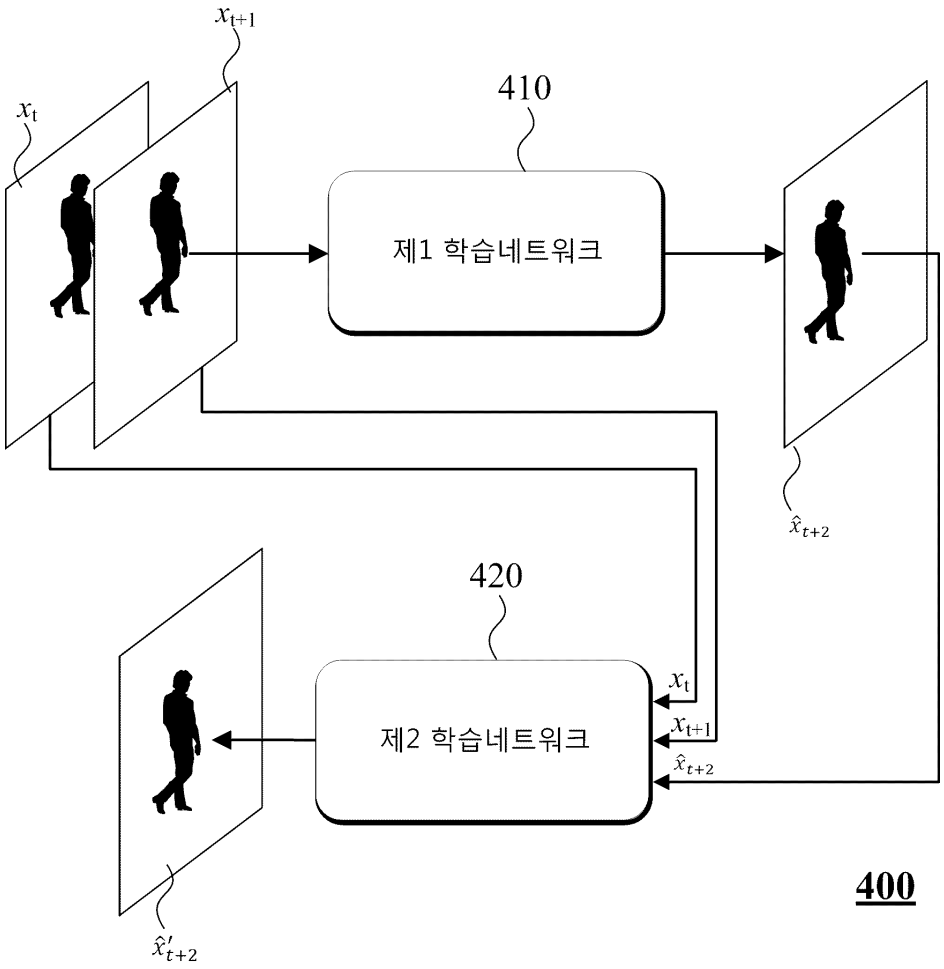

convolutional layer
pooling layer
211
x_{t-n}
x_{t+m}
s(t)
deconvolutional layer
unpooling layer
$\hat{x}_t$
212
x_{t+m}-x_{t-n}
r(t)
decoder(250)
Convolution LSTM(230)
encoder(210)
200

(A)

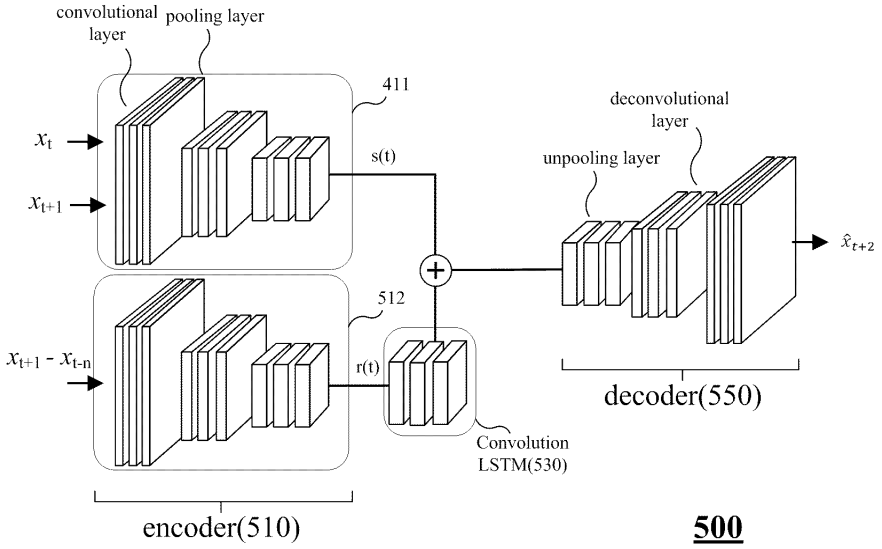

convolutional layer
pooling layer
311
x_{t-n}
$\hat{x}_t$
x_{t+m}
s(t)
deconvolutional layer
unpooling layer
$\hat{x}'_t$
312
x_{t+m}-x_{t-n}
r(t)
decoder(350)
Convolution LSTM(330)
encoder(310)
300

(B)

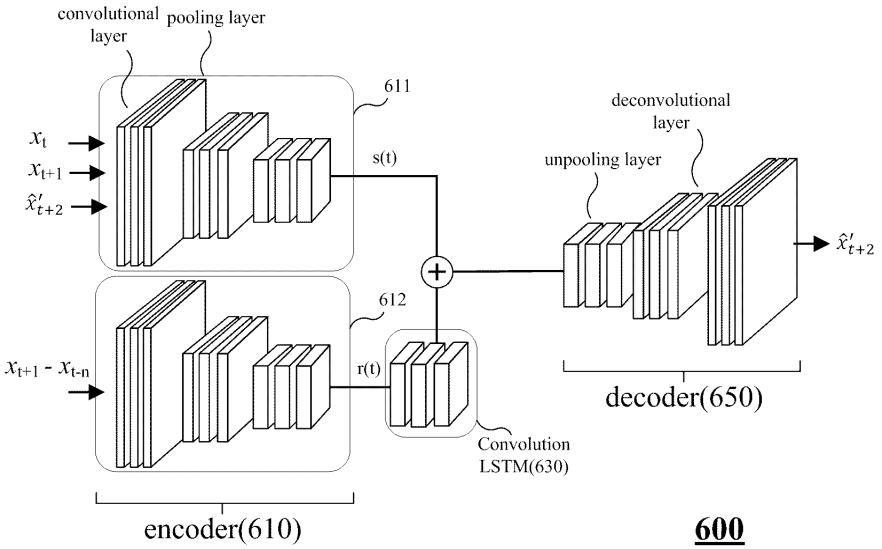
도면6


x_{t+1}
x_t
410
제1 학습네트워크
$\hat{x}_{t+2}$
420
제2 학습네트워크
x_t
x_{t+1}
$\hat{x}_{t+2}$
$\hat{x}'_{t+2}$
400

도면7


convolutional layer
pooling layer
411
x_t
x_{t+1}
s(t)
deconvolutional layer
unpooling layer
$\hat{x}_{t+2}$
512
$x_{t+1} - x_{t-n}$
r(t)
Convolution LSTM(530)
decoder(550)
encoder(510)
500

(A)


convolutional layer
pooling layer
611
x_t
x_{t+1}
$\hat{x}'_{t+2}$
s(t)
deconvolutional layer
unpooling layer
$\hat{x}'_{t+2}$
612
$x_{t+1} - x_{t-n}$
r(t)
Convolution LSTM(630)
decoder(650)
encoder(610)
600

(B)

도면8

