



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2023년03월17일
(11) 등록번호 10-2512018
(24) 등록일자 2023년03월15일

- (51) 국제특허분류(Int. Cl.)
G06T 3/00 (2019.01) G06N 3/02 (2023.01)
G06T 1/60 (2006.01) G06T 3/40 (2006.01)
G06T 7/11 (2017.01)
- (52) CPC특허분류
G06T 3/0056 (2013.01)
G06N 3/02 (2023.01)
- (21) 출원번호 10-2021-0168365
(22) 출원일자 2021년11월30일
심사청구일자 2021년11월30일
- (56) 선행기술조사문헌
Alexander H. Miller et al., 'Key-Value
Memory Networks for Directly Reading
Documents', arXiv:1606.03126v2 [cs.CL], 10
October 2016.*
Hsin-Ying Lee et al., 'Diverse
Image-to-Image Translation via Disentangled
Representations', Proceedings of the ECCV
2018, September 2018.*
*는 심사관에 의하여 인용된 문헌

- (73) 특허권자
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대
학교)
- (72) 발명자
손광훈
서울특별시 서대문구 연세로 50, 연세대학교 제3
공학관 C129호(신촌동)
정소미
서울특별시 서대문구 연세로 50, 연세대학교 제3
공학관 C129호(신촌동)
- (74) 대리인
민영준

전체 청구항 수 : 총 18 항

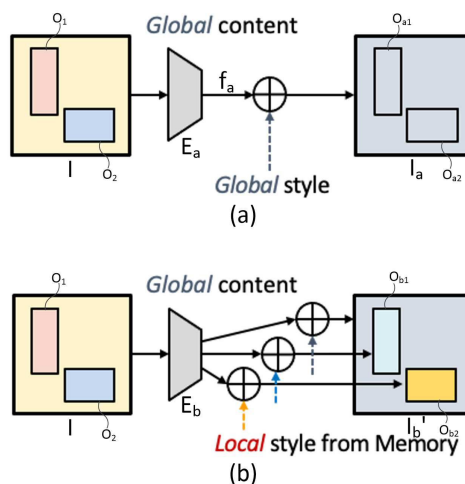
심사관 : 정성운

(54) 발명의 명칭 클래스 인지 메모리 네트워크 기반 영상 변환 장치 및 방법

(57) 요약

본 발명은 소스 도메인에서 획득된 입력 영상을 인가받아 신경망 연산하여, 입력 영상의 구조적 특징을 나타내는 콘텐츠 표현자를 획득하고, 입력 영상에 포함된 각 객체의 클래스에 따라 콘텐츠 표현자를 구분한 다수의 클래스 별 콘텐츠 표현자를 획득하는 인코딩부, 다수의 클래스 각각에 적어도 하나씩 대응하고, 각각 대응하는 클래스에 (뒷면에 계속)

대표도 - 도1



따른 객체의 구조적 대표 특성을 나타내는 하나의 아이템 키와 해당 클래스에서 소스 도메인과 상이한 타겟 도메인에서의 스타일적 대표 특성을 나타내는 스타일 값이 매칭되어 포함된 다수의 아이템이 저장된 메모리, 메모리에 저장된 다수의 아이템 중 클래스별 콘텐츠 표현자 각각에 대응하는 클래스의 아이템의 아이템 키들과의 유사도에 따른 리드 가중치를 계산하고, 계산된 리드 가중치를 아이템 키에 매칭된 스타일 값에 가중하여 타겟 스타일 표현자를 획득하는 타겟 스타일 생성부 및 콘텐츠 표현자와 타겟 스타일 표현자를 인가받아 신경망 연산하여 타겟 도메인에서의 출력 영상을 생성하는 출력 영상 생성부를 포함하여, 각 객체의 의미론적 특성이 잘 표현되면서 타겟 도메인에 대응하는 스타일로 변환되어 각 객체에 대한 의미론적 왜곡이 최소화된 자연스러운 타겟 도메인 영상을 획득할 수 있는 영상 변환 장치 및 방법을 제공한다.

(52) CPC특허분류

G06T 1/60 (2013.01)
G06T 3/4046 (2013.01)
G06T 7/11 (2017.01)
G06T 2207/20084 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711126179
과제번호	2016-0-00197-006
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송연구개발사업
연구과제명	스마트카 다중 센서와 딥러닝을 이용한 초정밀 내추럴 3D 뷰 생성 기술 개발
기 여 율	1/1
과제수행기관명	연세대학교 산학협력단
연구기간	2021.01.01 ~ 2021.12.31
공지예외적용	: 있음

명세서

청구범위

청구항 1

소스 도메인에서 획득된 입력 영상을 인가받아 신경망 연산하여, 상기 입력 영상의 구조적 특징을 나타내는 콘텐츠 표현자를 획득하고, 상기 입력 영상에 포함된 각 객체의 클래스에 따라 상기 콘텐츠 표현자를 구분한 다수의 클래스별 콘텐츠 표현자를 획득하는 인코딩부;

다수의 클래스 각각에 적어도 하나씩 대응하고, 각각 대응하는 클래스에 따른 객체의 구조적 대표 특성을 나타내는 하나의 아이템 키와 해당 클래스에서 상기 소스 도메인과 상이한 타겟 도메인에서의 스타일적 대표 특성을 나타내는 스타일 값이 매칭되어 포함된 다수의 아이템이 저장된 메모리;

상기 메모리에 저장된 다수의 아이템 중 상기 클래스별 콘텐츠 표현자 각각에 대응하는 클래스의 아이템의 아이템 키들과의 유사도에 따른 리드 가중치를 계산하고, 계산된 리드 가중치를 아이템 키에 매칭된 스타일 값에 가중하여 타겟 스타일 표현자를 획득하는 타겟 스타일 생성부; 및

상기 콘텐츠 표현자와 상기 타겟 스타일 표현자를 인가받아 신경망 연산하여 상기 타겟 도메인에서의 출력 영상을 생성하는 출력 영상 생성부를 포함하되,

상기 인코딩부는

미리 학습된 인공 신경망으로 구현되어 상기 입력 영상에 대해 신경망 연산을 수행하여 상기 입력 영상에 포함된 각 객체의 객체 영역과 클래스를 식별하는 객체 식별 표현자를 획득하는 객체 식별부;

미리 학습된 인공 신경망으로 구현되어 상기 입력 영상에 대해 신경망 연산하여, 상기 입력 영상의 구조적 특징을 나타내는 상기 콘텐츠 표현자를 획득하는 콘텐츠 표현자 획득부; 및

상기 객체 식별 표현자를 이용하여, 상기 콘텐츠 표현자에서 각 객체 영역을 클래스별로 구분하여 상기 다수의 클래스별 콘텐츠 표현자를 획득하는 콘텐츠 클러스터링부를 포함하는 영상 변환 장치.

청구항 2

삭제

청구항 3

제1항에 있어서, 상기 타겟 스타일 생성부는

상기 메모리에 저장된 다수의 아이템 중 다수의 클래스별 콘텐츠 각각에 대응하는 클래스의 아이템을 리드하고, 각 클래스별 콘텐츠 표현자에서 픽셀 단위의 픽셀 콘텐츠 표현자와 리드된 아이템 각각의 아이템 키 사이의 유사도를 기지정된 방식으로 계산하는 리드 유사도 계산부;

계산된 유사도에 따라 상기 픽셀 콘텐츠 표현자와 대응하는 클래스의 적어도 하나의 아이템 키 각각의 중요도를 나타내는 리드 가중치를 기지정된 방식으로 계산하는 리드 가중치 계산부;

대응하는 클래스의 적어도 하나의 아이템의 각 아이템 키에 매칭된 적어도 하나의 스타일 값에 대응하는 리드 가중치를 가중합하여, 픽셀 단위로 타겟 스타일을 나타내는 픽셀 타겟 스타일 표현자를 획득하고, 모든 클래스에 대한 픽셀 타겟 스타일 표현자를 해당 픽셀 위치에 배치하여 상기 타겟 스타일 표현자를 획득하는 타겟 스타일 표현자 획득부; 및

상기 콘텐츠 표현자와 상기 타겟 스타일 표현자를 결합하여 결합 타겟 스타일 표현자를 상기 출력 영상 생성부로 출력하는 타겟 스타일 결합부를 포함하는 영상 변환 장치.

청구항 4

제3항에 있어서, 상기 리드 유사도 계산부는

클래스별 콘텐츠 표현자(c_k^x)에서 각 픽셀 콘텐츠 표현자($c_p^x \in \{c_1^x, c_2^x, \dots, c_{N_k}^x\}$)와 해당 클래스에 할당된 아이템 키($k_n \in \{k_1, k_2, \dots, k_{M_k}\}$) 사이의 유사도($d(c_p^x, k_n)$)를 수학식

$$d(c_p^x, k_n) = \frac{c_p^x k_n^T}{\|c_p^x\|_2 \|k_n\|_2}$$

에 따라 계산하고,

상기 리드 가중치 계산부는

N_k 개의 픽셀 콘텐츠 표현자(c_p^x)에 대해 각각 계산된 M_k 개의 유사도($d(c_p^x, k_n)$)를 이용하여 클래스별 콘텐츠 표현자(c_k^x)의 클래스에 대응하는 M_k 개의 아이템 키(k_n) 각각에 대한 리드 가중치($\alpha_{p,n}^x$)를 수학식

$$\alpha_{p,n}^x = \frac{\exp(d(c_p^x, k_n))}{\sum_{n'=1}^{M_k} \exp(d(c_p^x, k_{n'}))}$$

으로 계산하며,

상기 타겟 스타일 표현자 획득부는

리드된 M_k 개의 스타일 값(v_n^y) 각각에 대응하는 M_k 개의 리드 가중치($\alpha_{p,n}^x$)를 기반으로 해당 클래스에서 각 픽셀에 대한 픽셀 타겟 스타일 표현자(\hat{s}_p^y)를 수학식

$$\hat{s}_p^y = \sum_{n'=1}^{M_k} \alpha_{p,n'}^x v_{n'}^y$$

에 따라 계산하는 영상 변환 장치.

청구항 5

제4항에 있어서, 상기 인코딩부는

학습 시에 상기 소스 도메인과 상기 타겟 도메인에서 획득된 입력 영상 중 적어도 하나를 인가받아, 인가된 적어도 하나의 입력 영상 각각에 대해 신경망 연산하여, 상기 적어도 하나의 입력 영상 각각의 소스 도메인의 스타일 특징을 나타내는 스타일 표현자를 획득하고, 상기 객체 식별 표현자를 이용하여, 상기 스타일 표현자에서 각 객체 영역을 클래스별로 구분하여 다수의 클래스별 스타일 표현자를 획득하는 클래스 스타일 표현자 추출부를 더 포함하고,

상기 영상 변환 장치는

학습시에 인가된 적어도 하나의 입력 영상 각각에 대한 상기 다수의 클래스별 콘텐츠 표현자와 상기 다수의 클래스별 스타일 표현자를 이용하여 상기 메모리에 저장된 다수의 아이템 각각에 포함된 아이템 키와 스타일 값을 업데이트하는 업데이트부를 더 포함하는 영상 변환 장치.

청구항 6

제5항에 있어서, 상기 업데이트부는

상기 메모리에 저장된 다수의 아이템 중 인가된 적어도 하나의 입력 영상 각각에 대한 상기 다수의 클래스별 콘텐츠 표현자 각각에 대응하는 클래스의 아이템을 리드하고, 각 클래스별 콘텐츠 표현자의 픽셀 콘텐츠 표현자와

리드된 아이템 각각의 아이템 키 사이의 유사도를 기지정된 방식으로 계산하는 업데이트 유사도 계산부;

계산된 유사도에 따라 상기 픽셀 콘텐츠 표현자와 대응하는 클래스의 적어도 하나의 아이템 키 각각의 중요도를 나타내는 업데이트 가중치를 기지정된 방식으로 계산하는 업데이트 가중치 계산부; 및

상기 픽셀 콘텐츠 표현자에 상기 업데이트 가중치를 가중합하고, 대응하는 클래스의 적어도 하나의 아이템의 각 아이템 키에 가산하여 업데이트된 아이템 키를 계산하고, 상기 픽셀 타겟 스타일 표현자에 상기 업데이트 가중치를 가중합하고, 대응하는 스타일 값에 가산하여 업데이트된 스타일 값을 계산하는 업데이트 값 계산부를 포함하는 영상 변환 장치.

청구항 7

제6항에 있어서, 상기 업데이트 유사도 계산부는

서로 다른 도메인의 입력 영상(I^x , I^y)에 각각 대응하는 픽셀 콘텐츠 표현자(c_p^x , c_p^y)와 해당 클래스에 할당된 아이템 키(k_n) 각각 사이의 유사도($d(c_p^x, k_n)$, $d(c_p^y, k_n)$)를 수학식

$$d(c_p^x, k_n) = \frac{c_p^x k_n^T}{\|c_p^x\|_2 \|k_n\|_2}$$

$$d(c_p^y, k_n) = \frac{c_p^y k_n^T}{\|c_p^y\|_2 \|k_n\|_2}$$

에 따라 계산하고,

상기 업데이트 가중치 계산부는

각 도메인에서 아이템 키(k_n) 각각에 대한 업데이트 가중치($\beta_{p,n}^x$, $\beta_{p,n}^y$)를 수학식

$$\beta_{p,n}^x = \frac{\exp(d(c_p^x, k_n))}{\sum_{p'=1}^{N_k} \exp(d(c_{p'}^x, k_n))}$$

$$\beta_{p,n}^y = \frac{\exp(d(c_p^y, k_n))}{\sum_{p'=1}^{N_{k'}} \exp(d(c_{p'}^y, k_n))}$$

에 따라 계산하는 영상 변환 장치.

청구항 8

제7항에 있어서, 상기 업데이트 값 계산부는

대응하는 클래스의 M_k 개의 아이템 키(k_n) 각각에 대한 업데이트된 아이템 키(\hat{k}_n)를 수학식

$$\hat{k}_n = \left\| k_n + \sum_{p'=1}^{N_k} \beta_{p',n}^x c_{p'}^x + \sum_{p'=1}^{N_{k'}} \beta_{p',n}^y c_{p'}^y \right\|_2$$

에 따라 계산하고,

상기 픽셀 타겟 스타일 표현자(s_p^x , s_p^y)에 대응하는 업데이트 가중치($\beta_{p,n}^x$, $\beta_{p,n}^y$)를 가중합하고, 메모리에서

리드된 대응하는 스타일 값(v_n^x, v_n^y)에 가산하여 업데이트된 스타일 값을 수학식

$$\hat{v}_n^x = \left\| v_n^x + \sum_{p'=1}^{N_k} \beta_{p',n}^x s_{p'}^x \right\|_2$$

$$\hat{v}_n^y = \left\| v_n^y + \sum_{p'=1}^{N_{k'}} \beta_{p',n}^y s_{p'}^y \right\|_2$$

에 따라 계산하는 영상 변환 장치.

청구항 9

제5항에 있어서, 상기 영상 변환 장치는

학습 시에 서로 다른 도메인의 입력 영상(I^x, I^y)에 각각 대응하는 상기 픽셀 콘텐츠 표현자(c_p^x, c_p^y)와 상기 픽셀 타겟 스타일 표현자(s_p^x, s_p^y)가 상기 메모리에 저장된 아이템 중 하나의 아이템에 대한 아이템 키(k_{p+}) 및 스타일 값(v_{p+}^x, v_{p+}^y)과는 차이가 작아지는 반면, 나머지 아이템에 대한 아이템 키(k_n) 및 스타일 값(v_n^x, v_n^y)과는 차이가 증가되도록 하는 키 손실(L_k) 및 스타일 손실(L_v)을 계산하고, 인공 신경망으로 구현되는 출력 영상 생성부를 학습시키기 위한 자기 재구성 손실(L^{self})과 순환 재구성 손실(L^{cyc})을 계산하며, 상기 키 손실(L_k)과 상기 스타일 손실(L_v), 상기 자기 재구성 손실(L^{self}) 및 상기 순환 재구성 손실(L^{cyc})을 기지정된 방식으로 가중합하여 총 손실을 계산하고, 계산된 상기 총 손실을 역전파하여 학습을 수행하는 학습부를 더 포함하는 영상 변환 장치.

청구항 10

제9항에 있어서, 상기 학습부는

상기 키 손실(L_k)을 수학식

$$\mathcal{L}_k = - \sum_{p=1}^N \log \frac{\exp(c_p^x \cdot k_{p+}/\tau)}{\sum_{n=1}^{M-1} \exp(c_p^x \cdot k_n/\tau)} - \sum_{p=1}^N \log \frac{\exp(c_p^y \cdot k_{p+}/\tau)}{\sum_{n=1}^{M-1} \exp(c_p^y \cdot k_n/\tau)}$$

에 따라 계산하고,

상기 스타일 손실(L_v)을 수학식

$$\mathcal{L}_v = - \sum_{p=1}^N \log \frac{\exp(s_p^x \cdot v_{p+}^x/\tau)}{\sum_{n=1}^{M-1} \exp(s_p^x \cdot v_n^x/\tau)} - \sum_{p=1}^N \log \frac{\exp(s_p^y \cdot v_{p+}^y/\tau)}{\sum_{n=1}^{M-1} \exp(s_p^y \cdot v_n^y/\tau)}$$

(여기서 k_{p+} 와 v_{p+}^x 및 v_{p+}^y 는 메모리에 저장된 M개의 아이템 각각의 아이템 키(k_n)와 스타일 값(v_n^x, v_n^y) 중 콘텐츠 표현자(c_p^x, c_p^y)와 픽셀 타겟 스타일 표현자(s_p^x, s_p^y)에 가장 유사한 아이템 키와 스타일 값을 나타내는 포지티브 샘플을 나타낸다.)

에 따라 계산하며,

상기 자기 재구성 손실을 동일한 도메인에 대한 콘텐츠 표현자(c^x, c^y)와 타겟 스타일 표현자(\hat{s}^x, \hat{s}^y)로부터 생

성되는 자기 재구성 영상($G^x(c^x, \hat{s}^x)$, $G^y(c^y, \hat{s}^y)$)과 입력 영상(I^x , I^y) 사이의 차이로 수학식

$$\mathcal{L}^{self} = \|G^x(c^x, \hat{s}^x) - I^x\|_1 + \|G^y(c^y, \hat{s}^y) - I^y\|_1$$

여기서 $G^x()$, $G^y()$ 는 각각 출력 영상 생성부의 신경망 연산을 표현한 함수이다.

으로 계산하고,

상기 순환 재구성 손실을 입력 영상(I^x , I^y)에서 변환된 영상(\hat{I}^y , \hat{I}^x)을 다시 입력 영상으로서 변환하여 획득되는 순환 재구성 영상($G^x(\hat{c}^y, \hat{s}^x)$, $G^y(\hat{c}^x, \hat{s}^y)$)과 입력 영상(I^x , I^y) 사이의 차이로 수학식

$$\mathcal{L}^{cyc} = \|G^x(\hat{c}^y, \hat{s}^x) - I^x\|_1 + \|G^y(\hat{c}^x, \hat{s}^y) - I^y\|_1$$

에 따라 계산하는 영상 변환 장치.

청구항 11

소스 도메인에서 획득된 입력 영상을 인가받아 신경망 연산하여, 상기 입력 영상의 구조적 특징을 나타내는 콘텐츠 표현자를 획득하고, 상기 입력 영상에 포함된 각 객체의 클래스에 따라 상기 콘텐츠 표현자를 구분한 다수의 클래스별 콘텐츠 표현자를 획득하는 단계;

다수의 클래스 각각에 적어도 하나씩 대응하여 메모리에 미리 저장되고 각각 대응하는 클래스에 따른 객체의 구조적 대표 특성을 나타내는 하나의 아이템 키와 해당 클래스에서 상기 소스 도메인과 상이한 타겟 도메인에서의 스타일적 대표 특성을 나타내는 스타일 값이 매칭되어 포함된 다수의 아이템 중 상기 클래스별 콘텐츠 표현자 각각에 대응하는 클래스의 아이템의 아이템 키들과의 유사도에 따른 리드 가중치를 계산하고, 계산된 리드 가중치를 아이템 키에 매칭된 스타일 값에 가중하여 타겟 스타일 표현자를 획득하는 단계; 및

상기 콘텐츠 표현자와 상기 타겟 스타일 표현자를 인가받아 신경망 연산하여 상기 타겟 도메인에서의 출력 영상을 생성하는 단계를 포함하되,

상기 클래스별 콘텐츠 표현자를 획득하는 단계는

미리 학습된 인공 신경망으로 구현되어 상기 입력 영상에 대해 신경망 연산을 수행하여 상기 입력 영상에 포함된 각 객체의 객체 영역과 클래스를 식별하는 객체 식별 표현자를 획득하는 단계;

미리 학습된 인공 신경망으로 구현되어 상기 입력 영상에 대해 신경망 연산하여, 상기 입력 영상의 구조적 특징을 나타내는 상기 콘텐츠 표현자를 획득하는 단계; 및

상기 객체 식별 표현자를 이용하여, 상기 콘텐츠 표현자에서 각 객체 영역을 클래스별로 구분하여 상기 다수의 클래스별 콘텐츠 표현자를 획득하는 단계를 포함하는 영상 변환 방법.

청구항 12

삭제

청구항 13

제11항에 있어서, 상기 타겟 스타일 표현자를 획득하는 단계는

상기 메모리에 저장된 다수의 아이템 중 다수의 클래스별 콘텐츠 각각에 대응하는 클래스의 아이템을 리드하고, 각 클래스별 콘텐츠 표현자에서 픽셀 단위의 픽셀 콘텐츠 표현자와 리드된 아이템 각각의 아이템 키 사이의 리드 유사도를 계산하는 단계;

계산된 리드 유사도에 따라 상기 픽셀 콘텐츠 표현자와 대응하는 클래스의 적어도 하나의 아이템 키 각각의 중요도를 나타내는 리드 가중치를 계산하는 단계;

대응하는 클래스의 적어도 하나의 아이템의 각 아이템 키에 매칭된 적어도 하나의 스타일 값에 대응하는 리드

가중치를 가중합하여, 픽셀 단위로 타겟 스타일을 나타내는 픽셀 타겟 스타일 표현자를 획득하고, 모든 클래스에 대한 픽셀 타겟 스타일 표현자를 해당 픽셀 위치에 배치하여 상기 타겟 스타일 표현자를 획득하는 단계; 및 상기 콘텐츠 표현자와 상기 타겟 스타일 표현자를 결합하여 결합 타겟 스타일 표현자를 출력하는 단계를 포함하는 영상 변환 방법.

청구항 14

제13항에 있어서, 상기 리드 유사도를 계산하는 단계는

클래스별 콘텐츠 표현자(c_k^x)에서 각 픽셀 콘텐츠 표현자($c_p^x \in \{c_1^x, c_2^x, \dots, c_{N_k}^x\}$)와 해당 클래스에 할당된 아이템 키($k_n \in \{k_1, k_2, \dots, k_{M_k}\}$) 사이의 리드 유사도($d(c_p^x, k_n)$)를 수학식

$$d(c_p^x, k_n) = \frac{c_p^x k_n^T}{\|c_p^x\|_2 \|k_n\|_2}$$

에 따라 계산하고,

상기 리드 가중치를 계산하는 단계는

N_k 개의 픽셀 콘텐츠 표현자(c_p^x)에 대해 각각 계산된 M_k 개의 리드 유사도($d(c_p^x, k_n)$)를 이용하여 클래스별 콘텐츠 표현자(c_k^x)의 클래스에 대응하는 M_k 개의 아이템 키(k_n) 각각에 대한 리드 가중치($\alpha_{p,n}^x$)를 수학식

$$\alpha_{p,n}^x = \frac{\exp(d(c_p^x, k_n))}{\sum_{n'=1}^{M_k} \exp(d(c_p^x, k_{n'}))}$$

으로 계산하며,

상기 타겟 스타일 표현자를 획득하는 단계는

리드된 M_k 개의 스타일 값(v_n^y) 각각에 대응하는 M_k 개의 리드 가중치($\alpha_{p,n}^x$)를 기반으로 해당 클래스에서 각 픽셀에 대한 픽셀 타겟 스타일 표현자(\hat{s}_p^y)를 수학식

$$\hat{s}_p^y = \sum_{n'=1}^{M_k} \alpha_{p,n'}^x v_{n'}^y$$

에 따라 계산하는 영상 변환 방법.

청구항 15

제14항에 있어서, 상기 영상 변환 방법은

다수의 클래스별 콘텐츠 표현자를 획득하는 단계 이전, 학습 단계를 더 포함하고,

상기 학습 단계는

상기 소스 도메인과 상기 타겟 도메인에서 획득된 입력 영상 중 적어도 하나를 인가받아, 인가된 적어도 하나의 입력 영상 각각에 대해 신경망 연산하여, 상기 적어도 하나의 입력 영상 각각에 대한 다수의 클래스별 콘텐츠 표현자와 함께 소스 도메인의 스타일 특징을 나타내는 스타일 표현자를 획득하고, 상기 객체 식별 표현자를 이용하여, 상기 스타일 표현자에서 각 객체 영역을 클래스별로 구분하여 다수의 클래스별 스타일 표현자를 획득하는 단계;

상기 다수의 클래스별 콘텐츠 표현자와 메모리에 저장된 대응하는 스타일 값에 따라 상기 타겟 스타일 표현자를

획득하고, 상기 콘텐츠 표현자와 상기 타겟 스타일 표현자를 신경망 연산하여 상기 입력 영상과 상이한 도메인의 출력 영상을 생성하는 단계; 및

인가된 적어도 하나의 입력 영상 각각에 대한 상기 다수의 클래스별 콘텐츠 표현자와 상기 다수의 클래스별 스타일 표현자를 이용하여 상기 메모리에 저장된 다수의 아이템 각각에 포함된 아이템 키와 스타일 값을 업데이트하는 단계를 포함하는 영상 변환 방법.

청구항 16

제15항에 있어서, 상기 업데이트하는 단계는

상기 메모리에 저장된 다수의 아이템 중 인가된 적어도 하나의 입력 영상 각각에 대한 상기 다수의 클래스별 콘텐츠 표현자 각각에 대응하는 클래스의 아이템을 리드하고, 각 클래스별 콘텐츠 표현자의 픽셀 콘텐츠 표현자와 리드된 아이템 각각의 아이템 키 사이의 업데이트 유사도를 계산하는 단계;

계산된 업데이트 유사도에 따라 상기 픽셀 콘텐츠 표현자와 대응하는 클래스의 적어도 하나의 아이템 키 각각의 중요도를 나타내는 업데이트 가중치를 계산하는 단계; 및

상기 픽셀 콘텐츠 표현자에 상기 업데이트 가중치를 가중합하고, 대응하는 클래스의 적어도 하나의 아이템의 각 아이템 키에 가산하여 업데이트된 아이템 키를 계산하고, 상기 픽셀 타겟 스타일 표현자에 상기 업데이트 가중치를 가중합하고, 대응하는 스타일 값에 가산하여 업데이트된 스타일 값을 계산하여 업데이트된 아이템을 획득하는 단계를 포함하는 영상 변환 방법.

청구항 17

제16항에 있어서, 상기 업데이트 유사도를 계산하는 단계는

서로 다른 도메인의 입력 영상(I^x , I^y)에 각각 대응하는 픽셀 콘텐츠 표현자(c_p^x , c_p^y)와 해당 클래스에 할당된 아이템 키(k_n) 각각 사이의 업데이트 유사도($d(c_p^x, k_n)$, $d(c_p^y, k_n)$)를 수학적

$$d(c_p^x, k_n) = \frac{c_p^x k_n^T}{\|c_p^x\|_2 \|k_n\|_2}$$

$$d(c_p^y, k_n) = \frac{c_p^y k_n^T}{\|c_p^y\|_2 \|k_n\|_2}$$

에 따라 계산하고,

상기 업데이트 가중치를 계산하는 단계는

각 도메인에서 아이템 키(k_n) 각각에 대한 업데이트 가중치($\beta_{p,n}^x$, $\beta_{p,n}^y$)를 수학적

$$\beta_{p,n}^x = \frac{\exp(d(c_p^x, k_n))}{\sum_{p'=1}^{N_k} \exp(d(c_{p'}^x, k_n))}$$

$$\beta_{p,n}^y = \frac{\exp(d(c_p^y, k_n))}{\sum_{p'=1}^{N_{k'}} \exp(d(c_{p'}^y, k_n))}$$

에 따라 계산하는 영상 변환 방법.

청구항 18

제17항에 있어서, 상기 업데이트된 아이টে를 획득하는 단계는

대응하는 클래스의 M_k 개의 아이테 키(k_n) 각각에 대한 업데이트된 아이테 키(\hat{k}_n)를 수학식

$$\hat{k}_n = \left\| k_n + \sum_{p'=1}^{N_k} \beta_{p',n}^x c_{p'}^x + \sum_{p'=1}^{N_{k'}} \beta_{p',n}^y c_{p'}^y \right\|_2$$

에 따라 계산하는 단계; 및

상기 픽셀 타겟 스타일 표현자(s_p^x, s_p^y)에 대응하는 업데이트 가중치($\beta_{p,n}^x, \beta_{p,n}^y$)를 가중합하고, 메모리에서 리드된 대응하는 스타일 값(v_n^x, v_n^y)에 가산하여 업데이트된 스타일 값을 수학식

$$\hat{v}_n^x = \left\| v_n^x + \sum_{p'=1}^{N_k} \beta_{p',n}^x s_{p'}^x \right\|_2$$

$$\hat{v}_n^y = \left\| v_n^y + \sum_{p'=1}^{N_{k'}} \beta_{p',n}^y s_{p'}^y \right\|_2$$

에 따라 계산하는 단계를 포함하는 영상 변환 방법.

청구항 19

제15항에 있어서, 상기 학습 단계는

학습 시에 서로 다른 도메인의 입력 영상(I^x, I^y)에 각각 대응하는 상기 픽셀 콘텐츠 표현자(c_p^x, c_p^y)와 상기 픽셀 타겟 스타일 표현자(s_p^x, s_p^y)가 상기 메모리에 저장된 아이테 중 하나의 아이테에 대한 아이테 키(k_{p+}) 및 스타일 값(v_{p+}^x, v_{p+}^y)과는 차이가 작아지는 반면, 나머지 아이테에 대한 아이테 키(k_n) 및 스타일 값(v_n^x, v_n^y)과는 차이가 증가되도록 하는 키 손실(L_k) 및 스타일 손실(L_v)을 계산하고, 인공 신경망으로 구현되는 출력 영상 생성부를 학습시키기 위한 자기 재구성 손실(L^{self})과 순환 재구성 손실(L^{cyc})을 계산하는 단계; 및

상기 키 손실(L_k)과 상기 스타일 손실(L_v), 상기 자기 재구성 손실(L^{self}) 및 상기 순환 재구성 손실(L^{cyc})을 기지정된 방식으로 가중합하여 총 손실을 계산하고, 계산된 상기 총 손실을 역전파하는 단계를 더 포함하는 영상 변환 방법.

청구항 20

제19항에 있어서, 상기 손실을 계산하는 단계는

상기 키 손실(L_k)을 수학식

$$\mathcal{L}_k = - \sum_{p=1}^N \log \frac{\exp(c_p^x \cdot k_{p+}/\tau)}{\sum_{n=1}^{M-1} \exp(c_p^x \cdot k_n/\tau)} - \sum_{p=1}^N \log \frac{\exp(c_p^y \cdot k_{p+}/\tau)}{\sum_{n=1}^{M-1} \exp(c_p^y \cdot k_n/\tau)}$$

에 따라 계산하는 단계;

상기 스타일 손실(\mathcal{L}_v)을 수학식

$$\mathcal{L}_v = -\sum_{p=1}^N \log \frac{\exp(s_p^x \cdot v_{p+}^x / \tau)}{\sum_{n=1}^{M-1} \exp(s_p^x \cdot v_n^x / \tau)} - \sum_{p=1}^N \log \frac{\exp(s_p^y \cdot v_{p+}^y / \tau)}{\sum_{n=1}^{M-1} \exp(s_p^y \cdot v_n^y / \tau)}$$

(여기서 k_{p+}^x 와 v_{p+}^x 및 v_{p+}^y 는 메모리에 저장된 M개의 아이템 각각의 아이템 키(k_n)와 스타일 값(v_n^x , v_n^y) 중 콘텐츠 표현자(c_p^x , c_p^y)와 픽셀 타겟 스타일 표현자(s_p^x , s_p^y)에 가장 유사한 아이템 키와 스타일 값을 나타내는 포지티브 샘플을 나타낸다.)

에 따라 계산하는 단계;

상기 자기 재구성 손실을 동일한 도메인에 대한 콘텐츠 표현자(c^x , c^y)와 타겟 스타일 표현자(\hat{s}^x , \hat{s}^y)로부터 생성되는 자기 재구성 영상($G^x(c^x, \hat{s}^x)$, $G^y(c^y, \hat{s}^y)$)과 입력 영상(I^x , I^y) 사이의 차이로 수학식

$$\mathcal{L}^{self} = \|G^x(c^x, \hat{s}^x) - I^x\|_1 + \|G^y(c^y, \hat{s}^y) - I^y\|_1$$

여기서 $G^x()$, $G^y()$ 는 각각 출력 영상 생성부의 신경망 연산을 표현한 함수이다.

으로 계산하는 단계; 및

상기 순환 재구성 손실을 입력 영상(I^x , I^y)에서 변환된 영상(\hat{I}^y , \hat{I}^x)을 다시 입력 영상으로서 변환하여 획득되는 순환 재구성 영상($G^x(\hat{c}^y, \hat{s}^x)$, $G^y(\hat{c}^x, \hat{s}^y)$)과 입력 영상(I^x , I^y) 사이의 차이로 수학식

$$\mathcal{L}^{cyc} = \|G^x(\hat{c}^y, \hat{s}^x) - I^x\|_1 + \|G^y(\hat{c}^x, \hat{s}^y) - I^y\|_1$$

에 따라 계산하는 단계를 포함하는 영상 변환 방법.

발명의 설명

기술 분야

[0001] 본 발명은 영상 변환 장치 및 방법에 관한 것으로, 클래스 인지 메모리 네트워크 기반 영상 변환 장치 및 방법에 관한 것이다.

배경 기술

[0002] 영상 변환(image-to-image translation)은 특정 도메인의 영상을 요구되는 다른 도메인의 영상으로 변환하는 기술로서, 소스 도메인의 입력 영상을 타겟 도메인의 출력 영상으로 변환하는 것을 목표로 하는 생성 모델(generative model) 분야 기술이다. 여기서 도메인은 영상 획득 조건, 주변 환경 등 영상 변환이 적용되는 분야에 따라 다양하게 지정될 수 있다. 예로서 특정 카메라에 의해 촬영된 영상을 다른 카메라에서 촬영된 영상으로 변환하고자 하는 경우, 카메라의 특성이 도메인이 될 수 있다. 즉 흑백 영상이나 적외선 영상 등을 RGB 영상으로 변환하는 경우에 영상 변환 기술이 적용될 수 있다. 그리고 주간에 촬영된 주간 영상을 야간에 촬영한 야간 영상으로 변환하고자 하는 경우나, 맑은 날씨에서 촬영된 영상을 흐린 날씨나 비오는 날씨에서 촬영된 영상으로 변환하고자 하는 경우 주변 환경이 도메인이 될 수 있다. 또한 테두리만 주어진 라벨 영상으로 실제 제품과 같은 영상을 생성하기 위해서도 이용될 수도 있다.

[0003] 이와 같은 영상 변환 기술은 다양한 분야에 이용될 수 있으나, 최근에는 대량의 학습 데이터를 요구하는 딥 러닝 분야에서 특히 영상 변환 기술을 필요로 하고 있다. 딥 러닝 모델은 학습 데이터가 부족한 경우, 요구되는 성능을 나타내지 못한다는 것은 잘 알려져 있으나, 다양한 도메인에서 이용되는 여러 딥 러닝 모델 각각에 대응하는 도메인의 학습 데이터를 대량으로 획득하는 것은 매우 어렵다. 이러한 경우에 영상 변환 기법은 이미 다른 도메인에서 획득된 대량의 영상 데이터를 딥 러닝 모델에 적합한 도메인의 학습 데이터로 변환시킬 수 있어,

학습 데이터 부족 문제를 용이하게 해소할 수 있다.

- [0004] 다만 기존의 영상 변환 기법은 소스 도메인에서 획득된 영상 내에 포함된 각 객체의 클래스에 따른 특성을 고려하지 않고, 영상 전체의 스타일을 타겟 도메인에 대응하는 스타일로 전환시켜 영상을 변환하는 방식을 이용함에 따라 영상 내에 포함된 각 객체에 대한 의미론적 왜곡이 발생하여 자연스럽지 않은 타겟 도메인 영상이 획득된다는 문제가 있었다. 그리고 이와 같은 의미론적 왜곡은 딥 러닝 모델의 학습 성능을 저하시켜 경우에 따라서는 객체 인식 오류를 유발하는 요인이 될 수도 있다.

선행기술문헌

특허문헌

- [0005] (특허문헌 0001) 한국 등록 특허 제10-2229572호 (2021.03.12 등록)

발명의 내용

해결하려는 과제

- [0006] 본 발명의 목적은 변환되는 영상에 포함된 각 객체의 의미론적 왜곡을 최소화하여 자연스러운 타겟 도메인 영상을 획득할 수 있는 영상 변환 장치 및 방법을 제공하는데 있다.
- [0007] 본 발명의 다른 목적은 영상에 포함된 각 객체의 클래스에 따라 메모리에 클래스별로 구분되어 미리 저장된 다수의 로컬 스타일 중 대응하는 로컬 스타일을 해당 객체에 적용하여 변환함으로써, 영상 내 각 객체의 특성을 유지하면서 타겟 도메인에 대응하는 스타일의 영상을 획득할 수 있는 영상 변환 장치 및 방법을 제공하는데 있다.

과제의 해결 수단

- [0008] 상기 목적을 달성하기 위한 본 발명의 일 실시예에 따른 영상 변환 장치는 소스 도메인에서 획득된 입력 영상을 인가받아 신경망 연산하여, 상기 입력 영상의 구조적 특징을 나타내는 콘텐츠 표현자를 획득하고, 상기 입력 영상에 포함된 각 객체의 클래스에 따라 상기 콘텐츠 표현자를 구분한 다수의 클래스별 콘텐츠 표현자를 획득하는 인코딩부; 다수의 클래스 각각에 적어도 하나씩 대응하고, 각각 대응하는 클래스에 따른 객체의 구조적 대표 특성을 나타내는 하나의 아이템 키와 해당 클래스에서 상기 소스 도메인과 상이한 타겟 도메인에서의 스타일적 대표 특성을 나타내는 스타일 값이 매칭되어 포함된 다수의 아이템이 저장된 메모리; 상기 메모리에 저장된 다수의 아이템 중 상기 클래스별 콘텐츠 표현자 각각에 대응하는 클래스의 아이템의 아이템 키들과의 유사도에 따른 리드 가중치를 계산하고, 계산된 리드 가중치를 아이템 키에 매칭된 스타일 값에 가중하여 타겟 스타일 표현자를 획득하는 타겟 스타일 생성부; 및 상기 콘텐츠 표현자와 상기 타겟 스타일 표현자를 인가받아 신경망 연산하여 상기 타겟 도메인에서의 출력 영상을 생성하는 출력 영상 생성부를 포함한다.
- [0009] 상기 인코딩부는 미리 학습된 인공 신경망으로 구현되어 상기 입력 영상에 대해 신경망 연산을 수행하여 상기 입력 영상에 포함된 각 객체의 객체 영역과 클래스를 식별하는 객체 식별 표현자를 획득하는 객체 식별부; 미리 학습된 인공 신경망으로 구현되어 상기 입력 영상에 대해 신경망 연산하여, 상기 입력 영상의 구조적 특징을 나타내는 상기 콘텐츠 표현자를 획득하는 콘텐츠 표현자 획득부; 및 상기 객체 식별 표현자를 이용하여, 상기 콘텐츠 표현자에서 각 객체 영역을 클래스별로 구분하여 상기 다수의 클래스별 콘텐츠 표현자를 획득하는 콘텐츠 클러스터링부를 포함할 수 있다.
- [0010] 상기 타겟 스타일 생성부는 상기 메모리에 저장된 다수의 아이템 중 다수의 클래스별 콘텐츠 각각에 대응하는 클래스의 아이템을 리드하고, 각 클래스별 콘텐츠 표현자에서 픽셀 단위의 픽셀 콘텐츠 표현자와 리드된 아이템 각각의 아이템 키 사이의 유사도를 기지정된 방식으로 계산하는 리드 유사도 계산부; 계산된 유사도에 따라 상기 픽셀 콘텐츠 표현자와 대응하는 클래스의 적어도 하나의 아이템 키 각각의 중요도를 나타내는 리드 가중치를 기지정된 방식으로 계산하는 리드 가중치 계산부; 대응하는 클래스의 적어도 하나의 아이템의 각 아이템 키에 매칭된 적어도 하나의 스타일 값에 대응하는 리드 가중치를 가중함으로써, 픽셀 단위로 타겟 스타일을 나타내는 픽셀 타겟 스타일 표현자를 획득하고, 모든 클래스에 대한 픽셀 타겟 스타일 표현자를 해당 픽셀 위치에 배치하여 상기 타겟 스타일 표현자를 획득하는 타겟 스타일 표현자 획득부; 및 상기 콘텐츠 표현자와 상기 타겟 스타일 표현자를 결합하여 결합 타겟 스타일 표현자를 상기 출력 영상 생성부로 출력하는 타겟 스타일 결합부를 포

함할 수 있다.

- [0011] 상기 인코딩부는 학습 시에 상기 소스 도메인과 상기 타겟 도메인에서 획득된 입력 영상 중 적어도 하나를 인가받아, 인가된 적어도 하나의 입력 영상 각각에 대해 신경망 연산하여, 상기 적어도 하나의 입력 영상 각각의 소스 도메인의 스타일 특징을 나타내는 스타일 표현자를 획득하고, 상기 객체 식별 표현자를 이용하여, 상기 스타일 표현자에서 각 객체 영역을 클래스별로 구분하여 다수의 클래스별 스타일 표현자를 획득하는 클래스 스타일 표현자 추출부를 더 포함할 수 있다.
- [0012] 상기 영상 변환 장치는 학습시에 인가된 적어도 하나의 입력 영상 각각에 대한 상기 다수의 클래스별 콘텐츠 표현자와 상기 다수의 클래스별 스타일 표현자를 이용하여 상기 메모리에 저장된 다수의 아이템 각각에 포함된 아이템 키와 스타일 값을 업데이트하는 업데이트부를 더 포함할 수 있다.
- [0013] 상기 업데이트부는 상기 메모리에 저장된 다수의 아이템 중 인가된 적어도 하나의 입력 영상 각각에 대한 상기 다수의 클래스별 콘텐츠 표현자 각각에 대응하는 클래스의 아이템을 리드하고, 각 클래스별 콘텐츠 표현자의 픽셀 콘텐츠 표현자와 리드된 아이템 각각의 아이템 키 사이의 유사도를 기지정된 방식으로 계산하는 업데이트 유사도 계산부; 계산된 유사도에 따라 상기 픽셀 콘텐츠 표현자와 대응하는 클래스의 적어도 하나의 아이템 키 각각의 중요도를 나타내는 업데이트 가중치를 기지정된 방식으로 계산하는 업데이트 가중치 계산부; 및 상기 픽셀 콘텐츠 표현자에 상기 업데이트 가중치를 가중합하고, 대응하는 클래스의 적어도 하나의 아이템의 각 아이템 키에 가산하여 업데이트된 아이템 키를 계산하고, 상기 픽셀 타겟 스타일 표현자에 상기 업데이트 가중치를 가중합하고, 대응하는 스타일 값에 가산하여 업데이트된 스타일 값을 계산하는 업데이트 값 계산부를 포함할 수 있다.
- [0014] 상기 영상 변환 장치는 학습 시에 서로 다른 도메인의 입력 영상(I^x, I^y)에 각각 대응하는 상기 픽셀 콘텐츠 표현자(c_p^x, c_p^y)와 상기 픽셀 타겟 스타일 표현자(s_p^x, s_p^y)가 상기 메모리에 저장된 아이템 중 하나의 아이템에 대한 아이템 키(k_{pt}) 및 스타일 값(v_{pt}^x, v_{pt}^y)과는 차이가 작아지는 반면, 나머지 아이템에 대한 아이템 키(k_n) 및 스타일 값(v_n^x, v_n^y)과는 차이가 증가되도록 하는 키 손실(L_k) 및 스타일 손실(L_v)을 계산하고, 인공 신경망으로 구현되는 출력 영상 생성부를 학습시키기 위한 자기 재구성 손실(L^{self})과 순환 재구성 손실(L^{cyc})을 계산하며, 상기 키 손실(L_k)과 상기 스타일 손실(L_v), 상기 자기 재구성 손실(L^{self}) 및 상기 순환 재구성 손실(L^{cyc})을 기지정된 방식으로 가중합하여 총 손실을 계산하고, 계산된 상기 총 손실을 역전파하여 학습을 수행하는 학습부를 더 포함할 수 있다.
- [0015] 상기 목적을 달성하기 위한 본 발명의 다른 실시예에 따른 영상 변환 방법은 소스 도메인에서 획득된 입력 영상을 인가받아 신경망 연산하여, 상기 입력 영상의 구조적 특징을 나타내는 콘텐츠 표현자를 획득하고, 상기 입력 영상에 포함된 각 객체의 클래스에 따라 상기 콘텐츠 표현자를 구분한 다수의 클래스별 콘텐츠 표현자를 획득하는 단계; 다수의 클래스 각각에 적어도 하나씩 대응하여 메모리에 미리 저장되고 각각 대응하는 클래스에 따른 객체의 구조적 대표 특성을 나타내는 하나의 아이템 키와 해당 클래스에서 상기 소스 도메인과 상이한 타겟 도메인에서의 스타일적 대표 특성을 나타내는 스타일 값이 매칭되어 포함된 다수의 아이템 중 상기 클래스별 콘텐츠 표현자 각각에 대응하는 클래스의 아이템의 아이템 키들과의 유사도에 따른 리드 가중치를 계산하고, 계산된 리드 가중치를 아이템 키에 매칭된 스타일 값에 가중하여 타겟 스타일 표현자를 획득하는 단계; 및 상기 콘텐츠 표현자와 상기 타겟 스타일 표현자를 인가받아 신경망 연산하여 상기 타겟 도메인에서의 출력 영상을 생성하는 단계를 포함한다.

발명의 효과

- [0016] 따라서, 본 발명의 실시예에 따른 영상 변환 장치 및 방법은 소스 도메인에서 획득되어 입력된 영상에 포함된 각 객체의 클래스를 인식하고, 인식된 클래스에 따라 메모리에 클래스별로 구분되어 저장된 다수의 로컬 스타일 중 대응하는 로컬 스타일을 해당 객체에 적용하여 타겟 도메인의 영상으로 변환함으로써, 각 객체의 의미론적 특성이 잘 표현되면서 타겟 도메인에 대응하는 스타일로 변환된 영상을 획득할 수 있다. 그러므로 영상에 포함된 각 객체에 대한 의미론적 왜곡이 최소화되어 변환된 자연스러운 타겟 도메인 영상을 획득할 수 있다.

도면의 간단한 설명

- [0017] 도 1은 기존의 영상 변환 방식과 본 실시예에 따른 영상 변환 방식의 개념 차이를 설명하기 위한 도면이다.
- 도 2는 본 발명의 일 실시예에 따른 영상 변환 장치의 개략적 구조를 나타낸다.
- 도 3은 도 2의 영상 변환 장치의 동작을 설명하기 위한 도면이다.
- 도 4는 도 2의 제1 인코더의 상세 구성의 일 예를 나타낸다.
- 도 5는 도 4의 콘텐츠 클러스터링부의 동작을 설명하기 위한 도면이다.
- 도 6은 도 2의 제1 타겟 스타일 생성부의 상세 구성의 일 예를 나타낸다.
- 도 7은 타겟 스타일 생성부와 업데이트부의 동작을 설명하기 위한 도면이다.
- 도 8은 도 2의 업데이트부의 상세 구성의 일 예를 나타낸다.
- 도 9는 본 실시예의 영상 변환 장치와 기존의 영상 변환 장치의 성능을 비교한 결과를 나타낸다.
- 도 10은 본 발명의 일 실시예에 따른 영상 변환 방법을 나타낸다.
- 도 11은 도 10의 영상 변환 방법을 위한 학습 단계의 일 예를 나타낸다.

발명을 실시하기 위한 구체적인 내용

- [0018] 본 발명과 본 발명의 동작상의 이점 및 본 발명의 실시예에 의하여 달성되는 목적을 충분히 이해하기 위해서는 본 발명의 바람직한 실시예를 예시하는 첨부 도면 및 첨부 도면에 기재된 내용을 참조하여야만 한다.
- [0019] 이하, 첨부한 도면을 참조하여 본 발명의 바람직한 실시예를 설명함으로써, 본 발명을 상세히 설명한다. 그러나, 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며, 설명하는 실시예에 한정되는 것이 아니다. 그리고, 본 발명을 명확하게 설명하기 위하여 설명과 관계없는 부분은 생략되며, 도면의 동일한 참조부호는 동일한 부재임을 나타낸다.
- [0020] 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라, 다른 구성요소를 더 포함할 수 있는 것을 의미한다. 또한, 명세서에 기재된 "...부", "...기", "모듈", "블록" 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어나 소프트웨어 또는 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다.
- [0021] 도 1은 기존의 영상 변환 방식과 본 실시예에 따른 영상 변환 방식의 개념 차이를 설명하기 위한 도면이다.
- [0022] 도 1에서 (a)는 기존의 영상 변환 방식을 나타내고, (b)는 본 실시예의 영상 변환 방식을 나타낸다. 우선 (a)를 살펴보면, 기존의 영상 변환 방식은 소스 도메인에서의 입력 영상(I)이 인가되면, 인코더(E_s)가 입력 영상(I)에 포함된 콘텐츠, 즉 객체(o_1, o_2)를 구분하지 않고, 입력 영상(I) 전체를 일괄적으로 인코딩하고, 인코딩된 특징맵(f_s)에 타겟 도메인의 전체적인 스타일을 나타내는 글로벌 스타일(Global style)을 적용하여 변환함으로써 출력 영상(I_a)을 획득한다. 이 경우, 입력 영상(I)에 포함된 객체(o_1, o_2)가 구분되지 않고 일괄적으로 글로벌 스타일이 적용됨에 따라, 변환된 출력 영상(I_a)에서는 입력 영상(I)에서와 달리 객체(o_{a1}, o_{a2})의 특성이 잘 나타나지 않게 된다. 예로서 입력 영상(I)의 소스 도메인이 주간이고, 출력 영상(I_a)의 타겟 도메인이 야간인 경우, 입력 영상(I)에 포함된 맑은 날씨에서의 건물, 도로, 차량, 나무, 하늘, 사람 등과 같이 서로 다른 특성을 갖고 있는 다양한 객체에 단일한 하나의 공통 스타일을 적용하여 변환하는 경우, 각 객체의 특성이 소실되어 의미론적 왜곡이 발생할 수 있다. 즉 타겟 도메인으로 획득된 출력 영상(I_a)에서 객체 식별이 용이하지 않을 수 있다.
- [0023] 그리고 이와 같은 출력 영상(I_a)에 발생된 의미론적 왜곡은 출력 영상(I_a)을 이용하여 학습된 장치, 예를 들면 자율 주행 장치 등이 주변 사물을 정상적으로 인지하지 못하여 사고를 유발하는 등의 문제를 야기할 수 있다.
- [0024] 반면 (b)에서는 입력 영상(I)이 인가되면, 인코더(E_b)가 입력 영상(I)에 포함된 콘텐츠, 즉 각 객체(o_1, o_2)의 클래스를 인지하여 구분하여 특징을 추출하고, 클래스에 따라 구분된 각 객체(o_1, o_2)의 영역에 대해 미리 클래스별로 획득된 서로 다른 로컬 스타일(Local style)을 적용하여 변환함으로써 출력 영상(I_b)을 획득한다. 즉

모든 객체(o_1, o_2)에 대해 대응하는 클래스에 따라 구분된 서로 다른 스타일을 적용하여 변환하므로, 의미론적 왜곡이 최소화되어 출력 영상(I_b)에 포함된 객체(o_{b1}, o_{b2}) 각각의 특성이 잘 나타나게 된다. 따라서 사실적인 자연스러운 출력 영상(I_b)을 획득할 수 있으며, 출력 영상(I_b)에 각 객체(o_{b1}, o_{b2})의 의미론적 특성이 잘 유지되어 있어, 이후 다른 딥 러닝 모델이 타겟 도메인의 출력 영상(I_b)을 학습 데이터로 이용하더라도 정확한 학습을 수행할 수 있다.

[0025] 도 2는 본 발명의 일 실시예에 따른 영상 변환 장치의 개략적 구조를 나타내고, 도 3은 도 2의 영상 변환 장치의 동작을 설명하기 위한 도면이다.

[0026] 본 실시예에 따른 영상 변환 장치는 실제 사용 시, 기지정된 소스 도메인의 단일 입력 영상을 인가받아 타겟 도메인의 출력 영상을 출력하도록 구성될 수 있다. 그러나 경우에 따라서는 서로 상이한 도메인의 두 영상 중 임의의 적어도 하나의 영상을 입력 영상으로 인가받아 상대 도메인의 영상으로 변환하도록 구성될 수도 있다. 이에 도 2에서는 2개의 서로 다른 도메인의 영상을 인가받아 각각 상대 도메인의 영상으로 변환할 수 있는 영상 변환 장치를 예로서 도시하였다.

[0027] 본 실시예에 따른 영상 변환 장치는 도 2에 도시된 바와 같이, 영상 획득부(100), 인코딩부(200), 타겟 스타일 생성부(300), 출력 영상 생성부(400) 및 메모리(500)를 포함할 수 있다.

[0028] 영상 획득부는 소스 도메인의 입력 영상(I^x, I^y)을 획득한다. 도 2에서 제1 영상 획득부(110)는 제1 소스 도메인의 제1 입력 영상(I^x)을 획득하고, 제2 영상 획득부(120)는 제2 소스 도메인의 제2 입력 영상(I^y)을 획득한다. 여기서 제1 소스 도메인은 제2 입력 영상(I^y)이 변환되는 제2 출력 영상(\hat{I}^y)의 타겟 도메인이고, 제2 소스 도메인은 제1 입력 영상(I^x)이 변환되는 제2 출력 영상(\hat{I}^x)의 타겟 도메인이다. 즉 영상 획득부는 서로 도메인이 상호 교차되어 획득되어야 하는 입력 영상(I^x, I^y)을 획득한다.

[0029] 일 예로 제1 입력 영상(I^x)은 야간 영상으로 변환되어야 하는 주간 영상일 수 있으며, 이 경우 제2 입력 영상(I^y)은 주간 영상으로 변환되어야 하는 야간 영상일 수 있다.

[0030] 인코딩부(200)는 미리 학습된 방식에 따라 영상 획득부에서 획득된 입력 영상(I^x, I^y)에 대해 신경망 연산을 수행하여, 입력 영상(I^x, I^y)에 포함된 각 객체의 클래스에 따른 구조적 특징을 나타내는 클래스별 콘텐츠 표현자(c_k^x, c_k^y)와 클래스에 따른 도메인 특징을 나타내는 클래스별 스타일 표현자(s_k^x, s_k^y)를 획득한다.

[0031] 구체적으로 인코딩부(200)는 입력 영상(I^x, I^y)의 구조적 특징을 나타내는 콘텐츠 표현자(c^x, c^y)와 도메인 특징을 나타내는 스타일 특징(s^x, s^y)을 획득하고, 입력 영상(I^x, I^y)에 포함된 각 객체의 객체 영역과 클래스를 식별하여, 클래스별 각 객체의 객체 영역에 대응하여 콘텐츠 표현자(c^x, c^y)와 스타일 표현자(s^x, s^y)를 분할하여 다수의 클래스별 콘텐츠 표현자(c_k^x, c_k^y)와 다수의 클래스별 스타일 표현자(s_k^x, s_k^y)를 획득한다.

[0032] 인코딩부(200)는 제1 입력 영상(I^x)으로부터 제1 클래스별 콘텐츠 표현자(c_k^x)와 제1 클래스별 스타일 표현자(s_k^x)를 획득하기 위한 제1 인코더(210)와 제2 입력 영상(I^y)으로부터 제1 클래스별 콘텐츠 표현자(c_k^y)와 제1 클래스별 스타일 표현자(s_k^y)를 획득하기 위한 제2 인코더(220)를 포함할 수 있다.

[0033] 여기서 제1 인코더(210)와 제2 인코더(220)는 서로 다른 소스 도메인에서 획득된 제1 입력 영상(I^x)과 제2 입력 영상(I^y)을 인가받는 점에서만 서로 상이할 뿐 동일한 구성으로 구현될 수 있다.

[0034] 도 4는 도 2의 제1 인코더의 상세 구성의 일 예를 나타내고, 도 5는 도 4의 콘텐츠 클러스터링부의 동작을 설명하기 위한 도면이다.

[0035] 상기한 바와 같이 제1 인코더(210)와 제2 인코더(220)는 동일한 구성을 가지므로, 도 4에서는 설명의 편의를 위

하여, 제1 입력 영상(I^x)을 인가받아 제1 클래스별 콘텐츠 표현자(c_k^x)와 제1 클래스별 스타일 표현자(s_k^x)를 획득하는 제1 인코더(210)의 구성만을 도시하였다. 이에 이하에서는 편의를 위하여 제1 인코더(210)와 제2 인코더(220)의 구성과 신호를 구분하기 위한 제1 및 제2 와 같은 식별 표현을 생략하여 설명한다.

[0036] 도 4를 참조하면, 제1 인코더(210)는 객체 식별부(211), 콘텐츠 표현자 획득부(212), 콘텐츠 클러스터링부(213), 스타일 표현자 획득부(214) 및 스타일 클러스터링부(215)를 포함할 수 있다.

[0037] 미리 학습된 인공 신경망으로 구현되는 객체 식별부(211)는 제1 입력 영상(I^x)을 인가받아 신경망 연산하여, 제1 입력 영상(I^x)에서 각 객체가 포함된 객체 영역을 판별하고, 각 객체의 클래스를 식별하여 객체 식별 표현자(f_k^x)를 획득한다. 여기서 객체 식별 표현자(f_k^x)는 제1 입력 영상(I^x)에서 객체의 클래스와 각 클래스별 객체가 포함되는 영역을 나타내는 경계 박스의 위치와 크기를 지정한다.

[0038] 콘텐츠 표현자 획득부(212)는 미리 학습된 인공 신경망으로 구현되어 제1 입력 영상(I^x)을 인가받아 신경망 연산하여, 제1 입력 영상(I^x)의 전체적인 구조적 특징을 나타내는 콘텐츠 표현자(c^x)를 획득한다.

[0039] 그리고 콘텐츠 클러스터링부(213)는 콘텐츠 표현자(c^x)에서 객체 식별 표현자(f_k^x)에 의해 지정된 각 클래스에 따른 객체 영역에 포함된 픽셀들을 구분하여 클러스터링함으로써, 식별되는 클래스의 개수(K)에 대응하는 개수의 제1 클래스별 콘텐츠 표현자(c_k^x)(여기서 $k \in \{1, \dots, K\}$)를 획득한다. 콘텐츠 클러스터링부(213)는 도 5에 도시된 바와 같이 콘텐츠 표현자(c^x)와 함께 각 클래스에 따른 객체 영역을 나타내는 객체 식별 표현자(f_k^x)가 인가되면, 콘텐츠 표현자(c^x) 내에서 객체 영역의 픽셀들을 K개의 클래스별로 구분하여 클러스터링(class-wise clustering)함으로써, K개의 제1 클래스별 콘텐츠 표현자(c_k^x)를 획득할 수 있다.

[0040] 한편, 스타일 표현자 획득부(214) 또한 미리 학습된 인공 신경망으로 구현되어 제1 입력 영상(I^x)을 인가받아 신경망 연산하여, 제1 입력 영상(I^x)의 전체적인 도메인 특징을 나타내는 스타일 표현자(s^x)를 획득한다.

[0041] 그리고 스타일 클러스터링부(215)는 스타일 표현자(s^x)에서 객체 식별 표현자(f_k^x)에 의해 지정된 각 클래스에 따른 객체 영역에 포함된 픽셀들을 구분하여 클러스터링함으로써, 식별되는 클래스의 개수(K)에 대응하는 개수의 제1 클래스별 스타일 표현자(s_k^x)를 획득한다.

[0042] K개의 제1 클래스별 콘텐츠 표현자(c_k^x)와 K개의 제1 클래스별 스타일 표현자(s_k^x) 각각은 콘텐츠 표현자(c^x)와 스타일 표현자(s^x)에서 각 클래스(k)에 해당하는 픽셀 수 $c_k^x \in R^{N_k \times d}$ 과 $s_k^x \in R^{N_k \times d}$ 의 차원으로 획득된다.

[0043] 여기서는 이해의 편의를 위하여 콘텐츠 표현자 획득부(212)와 콘텐츠 클러스터링부(213)를 구분하여 도시하였으며, 스타일 표현자 획득부(214)와 스타일 클러스터링부(215)를 구분하여 도시하였으나, 콘텐츠 표현자 획득부(212)와 콘텐츠 클러스터링부(213)는 클래스 콘텐츠 표현자 추출부로 통합될 수 있으며, 스타일 표현자 획득부(214)와 스타일 클러스터링부(215)는 클래스 스타일 표현자 추출부로 통합될 수 있다.

[0044] 또한 스타일 표현자 획득부(214)와 스타일 클러스터링부(215)는 영상 변환 장치의 메모리(500)에 저장되어야 하는 각 객체의 클래스에 따른 스타일을 학습 시에 추출하기 위한 구성으로 학습이 종료된 이후에는 생략될 수 있다.

[0045] 다시 도 2 및 도 3을 참조하면, 메모리(500)에는 K개의 클래스 각각에 적어도 하나씩 대응하는 다수의 아이템이 미리 저장되며, 다수의 아이템 각각에는 클래스에 따른 객체의 구조적 대표 특성을 나타내는 하나의 아이템 키(k)와 해당 클래스에서 서로 다른 도메인에서의 스타일적 대표 특성을 각각 나타내는 2개의 스타일 값(v^x, v^y)이 서로 매칭되어 포함된다.

[0046] 메모리(500)에는 M개의 아이템이 저장될 수 있으며, 각 클래스별로 M_k ($M = \sum_{k=1}^K M_k$)개의 아이템이 할당될

수 있다. 즉 K개의 클래스 각각에 대해 M_k 개의 아이템이 할당될 수 있다. 여기서 각 클래스에 할당되는 아이템의 개수(M_k)는 클래스에 따라 서로 상이하게 할당될 수도 있다. 예로서 도 3에서 제1 클래스(Class 1)에 대해서는 3개의 아이템이 할당된 반면, 제2 클래스(Class 2)와 제K 클래스(Class K)에 대해서는 각각 2개의 아이템이 할당되었음을 알 수 있다. 즉 메모리(500)에는 전체 M개의 아이템이 K개의 클래스 각각에 대해 적어도 하나씩 할당되어 저장될 수 있다.

[0047] M개의 아이템 각각에서 아이템 키(k)는 대응하는 클래스의 구조, 즉 대응하는 클래스에 따른 콘텐츠의 대표 특징을 나타내는 콘텐츠 대푯값이고, 2개의 스타일 값(v^x, v^y) 중 제1 스타일 값은 (v^x)은 대응하는 클래스의 제1 도메인, 즉 대응하는 클래스의 제1 스타일 대표 특징을 나타내는 스타일 대푯값이며, 제2 스타일 값은 (v^y)은 대응하는 클래스의 제2 스타일 대표 특징을 나타내는 스타일 대푯값이다.

[0048] 즉 메모리(500)에는 대응하는 클래스의 구조적 대표 특징을 나타내는 아이템 키(k)와 도메인에 따른 서로 다른 스타일 대표 특징을 나타내는 2개의 스타일 값(v^x, v^y)을 포함하는 M개의 아이템이 저장된다. 여기서 아이템 키(k)와 2개의 스타일 값(v^x, v^y) 각각은 콘텐츠 표현자(c^x, c^y)와 스타일 표현자(s^x, s^y)의 채널 길이(d)와 동일한 길이(d)를 갖는 1차원 벡터이다.

[0049] 본 실시예에서 각 클래스에 대해 M_k 개의 아이템을 할당하는 것은 클래스에 따라 하나의 구조적 대표 특징이나 하나의 스타일 대표 특징만으로 해당 클래스에 포함되는 다양한 객체를 모두 표현하기 어려운 경우가 빈번하게 발생하기 때문이다. 예로서 차량에 대한 클래스의 경우, 트럭, 버스, 승용차 등과 같이 서로 매우 상이한 형상을 갖는 객체가 동일한 차량 클래스로 분류될 수 있다. 비록 차량 클래스에 포함되는 객체의 형상이 서로 상이할지라도, 각 차량을 나타내는 객체가 동일 클래스로 식별되었다면, 이는 공통된 구조적 특징이 검출된 것을 의미하므로, 동일 도메인에서만 객체에 대한 처리를 수행한다면 크게 문제가 되지 않을 수 있다. 그러나 다른 도메인에서 공통된 구조적 특징이 검출되어 동일 클래스로 식별될 객체들일지라도 도메인이 변경되는 경우, 해당 특징이 정상적으로 도출되지 않아 동일 클래스의 객체로 식별되지 않을 수 있다.

[0050] 이러한 문제를 방지하기 위해, 본 실시예에서는 각 클래스에 대해 다수의 아이템이 할당될 수 있도록 하여, 동일 클래스에서도 서로 상이한 객체의 다양한 특징을 각각 대표할 수 있도록 한다. 여기서 각 클래스별로 할당되는 아이템의 개수는 각 클래스에 속하는 객체의 특성을 고려하여 미리 지정될 수 있다.

[0051] 그리고 메모리(500)에 저장되는 아이템에 포함된 아이템 키(k)와 2개의 스타일 값(v^x, v^y)은 후술하는 업데이트 부(600)에 의해 학습 시에 미리 획득되어 저장된다.

[0052] 타겟 스타일 생성부(300)는 인코딩부(200)로부터 클래스별 콘텐츠 표현자(c_k^x, c_k^y)를 인가받고, 메모리(500)에 저장된 다수의 아이템 중 인가된 클래스별 콘텐츠 표현자(c_k^x, c_k^y)에 대응하는 클래스의 아이템의 아이템 키(k)들과의 유사도에 따른 리드 가중치($a_{p,n}^x, a_{p,n}^y$)를 계산하고, 계산된 리드 가중치($a_{p,n}^x, a_{p,n}^y$)를 아이템 키(k)에 매칭된 타겟 스타일 값(v^y, v^x)에 가중하여 타겟 스타일 표현자(\hat{s}^y, \hat{s}^x)를 획득한다. 그리고 콘텐츠 표현자(c^x, c^y)와 획득된 타겟 스타일 표현자(\hat{s}^y, \hat{s}^x)를 결합하여, 결합 타겟 스타일 표현자($(c^x, \hat{s}^y), (c^y, \hat{s}^x)$)를 출력한다.

[0053] 타겟 스타일 생성부(300)는 제1 클래스별 콘텐츠 표현자(c_k^x)를 인가받아 제1 결합 타겟 스타일 표현자(c^x, \hat{s}^y)를 출력하는 제1 타겟 스타일 생성부(310)와 제2 클래스별 콘텐츠 표현자(c_k^y)를 인가받아 제2 결합 타겟 스타일 표현자(c^y, \hat{s}^x)를 출력하는 제2 타겟 스타일 생성부(320)를 포함할 수 있다.

[0054] 도 6은 도 2의 제1 타겟 스타일 생성부의 상세 구성의 일 예를 나타내고, 도 7은 타겟 스타일 생성부와 업데이트부의 동작을 설명하기 위한 도면이다.

[0055] 도 6에서도 설명의 편의를 위하여 제1 타겟 스타일 생성부(310)만을 도시하였으나, 제2 타겟 스타일 생성부(320)도 동일한 구조를 갖는다.

[0056] 도 6 및 도 7을 참조하면, 제1 타겟 스타일 생성부(310)는 리드 유사도 계산부(311), 리드 가중치 계산부(312), 타겟 스타일 표현자 획득부(313) 및 타겟 스타일 결합부(314)를 포함할 수 있다.

[0057] 리드 유사도 계산부(311)는 K개의 클래스별 콘텐츠 표현자(c_k^x)를 인가받고, 인가된 클래스별 콘텐츠 표현자(c_k^x)의 클래스에 따라 메모리(500)에 저장된 다수의 아이템 중 대응하는 클래스에 할당된 M_k 개의 아이템 각각에 포함된 아이템 키(k)를 리드한다. 이때, 리드 유사도 계산부(311)는 K개의 클래스별 콘텐츠 표현자(c_k^x) 각각에서 픽셀 단위의 픽셀 콘텐츠 표현자($c_{k,p}^x$)를 추출한다. 이하에서는 편의를 위해, 클래스를 식별하기 위한 아래 첨자(k)는 삭제하여 표기한다. 따라서 해당 클래스에서의 픽셀 개수(N_k)만큼의 픽셀 콘텐츠 표현자가 클래스별 콘텐츠 표현자($c_k^x = \{c_1^x, c_2^x, \dots, c_{N_k}^x\}$)에 포함된다. 리드 유사도 계산부(311)는 해당 클래스의 클래스별 콘텐츠 표현자(c_k^x)에서 각 픽셀 콘텐츠 표현자($c_p^x \in \{c_1^x, c_2^x, \dots, c_{N_k}^x\}$)와 해당 클래스에 할당된 아이템 키($k_n \in \{k_1, k_2, \dots, k_{M_k}\}$) 사이의 유사도($d(c_p^x, k_n)$)를 수학적 식 1에 따라 계산한다.

수학적 식 1

$$d(c_p^x, k_n) = \frac{c_p^x k_n^T}{\|c_p^x\|_2 \|k_n\|_2}$$

[0058]

[0059] 여기서 $\|\cdot\|$ 는 L2 놈 함수를 나타내고, 위첨자 T는 전치 행렬을 나타낸다.

[0060] 따라서 리드 유사도 계산부(311)는 각 클래스에 대해 $N_k \times M_k$ 개수의 유사도를 계산한다.

[0061] 리드 가중치 계산부(312)는 리드 유사도 계산부(311)에서 N_k 개의 픽셀 콘텐츠 표현자(c_p^x)에 대해 각각 계산된 M_k 개의 유사도($d(c_p^x, k_n)$)를 기반으로 해당 클래스에서 M_k 개의 아이템 키(k_n) 각각에 대한 리드 가중치($\alpha_{p,n}^x$)를 수학적 식 2에 따라 계산한다.

수학적 식 2

$$\alpha_{p,n}^x = \frac{\exp(d(c_p^x, k_n))}{\sum_{n'=1}^{M_k} \exp(d(c_p^x, k_{n'}))}$$

[0062]

[0063] 수학적 식 2에 따르면, 리드 가중치($\alpha_{p,n}^x$)는 픽셀 콘텐츠 표현자에 대응하는 클래스의 적어도 하나의 아이템 키 각각의 중요도를 유사도를 이용하여 표현한 값이다.

[0064] 클래스 내의 M_k 개의 아이템 키(k_n) 각각에 대한 M_k 개의 리드 가중치($\alpha_{p,n}^x$)가 계산되면, 타겟 스타일 표현자 획득부(313)는 메모리(500)에서 M_k 개의 아이템 키(k_n) 각각에 매칭된 2개의 스타일 값(v_n^x, v_n^y) 중 타겟 도메인에 대한 M_k 개의 스타일 값(여기서는 v_n^y)을 리드하고, 리드된 M_k 개의 스타일 값(v_n^y) 각각에 대응하는 M_k 개의 리드 가중치($\alpha_{p,n}^x$)를 수학적 식 3과 같이 가중합하여, 해당 클래스에서 각 픽셀에 대한 픽셀 타겟 스타일 표현자(\hat{s}_p^y)를 획득한다.

수학식 3

$$\hat{\mathbf{s}}_p^y = \sum_{n'=1}^{M_k} \alpha_{p,n'}^x \mathbf{v}_{n'}^y$$

[0065]

[0066]

그리고 타겟 스타일 결합부(314)는 다수의 클래스 각각에 대한 픽셀 단위의 픽셀 타겟 스타일 표현자($\hat{\mathbf{s}}_p^y$)가 획득되면, 모든 클래스의 픽셀 타겟 스타일 표현자($\hat{\mathbf{s}}_p^y$)를 해당 픽셀 위치에 배치하여 타겟 스타일 표현자($\hat{\mathbf{s}}^y$)를 획득하고, 타겟 스타일 표현자($\hat{\mathbf{s}}^y$)를 콘텐츠 표현자(\mathbf{c}^x)와 결합(concatenate)하여 제1 결합 타겟 스타일 표현자($\mathbf{c}^x, \hat{\mathbf{s}}^y$)를 출력한다.

[0067]

즉 타겟 스타일 생성부(300)는 소스 도메인의 입력 영상(\mathbf{I}^x)로부터 추출된 객체의 구조적 특징인 클래스별 콘텐츠 표현자(\mathbf{c}_k^x)들과 메모리에 미리 저장된 대응하는 클래스의 대표 구조적 특징인 아이템 키(\mathbf{k}_n)들 사이의 픽셀 단위의 유사도에 따른 가중치를 해당 클래스의 대표 타겟 스타일인 스타일 값(\mathbf{v}_n^y)에 가중함으로써, 소스 도메인의 객체의 구조적 특징에 대응하는 타겟 도메인의 스타일인 타겟 스타일 표현자($\hat{\mathbf{s}}^y$)를 획득한다.

[0068]

다시 도 2 및 도3을 참조하면, 출력 영상 생성부(400)는 결합 타겟 스타일 표현자($(\mathbf{c}^x, \hat{\mathbf{s}}^y), (\mathbf{c}^y, \hat{\mathbf{s}}^x)$)를 인가받아 타겟 도메인에 대응하는 스타일의 출력 영상($\hat{\mathbf{I}}^y, \hat{\mathbf{I}}^x$)을 생성한다.

[0069]

출력 영상 생성부(400)는 제1 출력 영상 생성부(410)와 제2 출력 영상 생성부(420)를 포함할 수 있다. 제1 출력 영상 생성부(410)는 미리 학습된 인공 신경망으로 구현되고, 제1 결합 타겟 스타일 표현자($\mathbf{c}^x, \hat{\mathbf{s}}^y$)를 인가받아 신경망 연산하여, 타겟 도메인인 제2 도메인에 대응하는 스타일의 제1 출력 영상($\hat{\mathbf{I}}^y$)을 생성한다.

[0070]

마찬가지로 제2 출력 영상 생성부(420) 또한 미리 학습된 인공 신경망으로 구현되어, 인가된 제2 결합 타겟 스타일 표현자($\mathbf{c}^y, \hat{\mathbf{s}}^x$)에 대해 신경망 연산하여, 타겟 도메인인 제1 도메인에 대응하는 스타일의 제2 출력 영상($\hat{\mathbf{I}}^x$)을 생성한다.

[0071]

결과적으로 도 2에 도시된 본 실시예에 따른 영상 변환 장치는 제1 도메인에서 획득된 제1 입력 영상(\mathbf{I}^x)으로부터 제2 도메인 스타일의 제1 출력 영상($\hat{\mathbf{I}}^y$)을 획득하고, 제2 도메인에서 획득된 제2 입력 영상(\mathbf{I}^y)으로부터 제1 도메인 스타일의 제2 출력 영상($\hat{\mathbf{I}}^x$)을 획득할 수 있다. 즉 소스 도메인에서 획득된 입력 영상($\mathbf{I}^x, \mathbf{I}^y$)을 타겟 도메인의 출력 영상($\hat{\mathbf{I}}^y, \hat{\mathbf{I}}^x$)으로 변환할 수 있다.

[0072]

특히 메모리(500)에 클래스별로 구분되어 미리 저장된 타겟 도메인의 스타일을 이용하여 입력 영상($\mathbf{I}^x, \mathbf{I}^y$)에 포함된 각 객체의 클래스에 따라 서로 다른 스타일을 적용하여 변환함으로써, 매우 사실적인 타겟 도메인의 출력 영상($\hat{\mathbf{I}}^y, \hat{\mathbf{I}}^x$)을 획득할 수 있다.

[0073]

상기한 바와 같이, 도 2에서는 서로 다른 도메인의 영상을 상호 변환 가능한 영상 변환 장치를 도시한 것으로서, 만일 특정 도메인의 영상을 다른 지정된 도메인의 영상으로 변환하고자 하는 경우, 제2 입력 영상(\mathbf{I}^y)으로부터 제2 출력 영상($\hat{\mathbf{I}}^x$)을 생성하기 위해서만 구비된 각 구성 요소는 모두 생략될 수 있다. 즉 제2 영상 획득부(120), 제2 인코더(220), 제2 타겟 스타일 생성부(320) 및 제2 출력 영상 생성부(420)는 생략될 수 있다.

[0074]

한편, 상기한 본 실시예의 영상 변환 장치는 다수의 인공 신경망을 포함하므로, 실제 이용하기 위해서는 미리 학습이 되어야 한다. 뿐만 아니라 본 실시예의 영상 변환 장치는 메모리에 저장된 다수의 아이템에 기반하여

각 클래스의 객체를 타겟 스타일로 변환하므로, 메모리(500)에 저장된 아이템이 학습에 의해 미리 업데이트되어야 한다. 이에 본 실시예의 영상 변환 장치는 메모리(500)에 저장된 다수의 아이템을 업데이트하기 위한 업데이트부(600)를 포함한다. 업데이트부(600)는 학습 이후에도 계속적으로 메모리(500)에 저장된 아이템을 업데이트할 수 있도록 유지될 수도 있다.

[0075] 업데이트부(600)는 타겟 스타일 생성부(300)와 유사하게 인코딩부(200)로부터 제1 및 제2 클래스별 콘텐츠 표현자(c_k^x, c_k^y) 중 적어도 하나를 인가받고, 메모리(500)에 저장된 다수의 아이템 중 인가된 클래스별 콘텐츠 표현자(c_k^x, c_k^y)에 대응하는 클래스의 아이템의 아이템 키(k_n)들과의 유사도에 따른 업데이트 가중치($\beta_{p,n}^x, \beta_{p,n}^y$)를 계산하고, 계산된 업데이트 가중치($\beta_{p,n}^x, \beta_{p,n}^y$)와 인코딩부(200)에서 인가되는 클래스별 스타일 표현자(s_k^x, s_k^y)를 기반으로 각 아이템 키(k_n) 및 아이템 키(k_n)에 매칭된 타겟 스타일 값(v_n^y, v_n^x)에 대한 업데이트 값($\hat{k}_n^x, \hat{v}_n^x, \hat{v}_n^y$)들을 획득한다.

[0076] 도 8은 도 2의 업데이트부의 상세 구성의 일 예를 나타낸다.

[0077] 도 7 및 도 8을 참조하면, 업데이트부(600)는 업데이트 유사도 계산부(611), 업데이트 가중치 계산부(612) 및 업데이트 값 계산부(613)를 포함할 수 있다.

[0078] 업데이트 유사도 계산부(611)는 리드 유사도 계산부(311)와 마찬가지로 K개의 제1 및 제2 클래스별 콘텐츠 표현자(c_k^x, c_k^y) 중 적어도 하나를 인가받고, 인가된 클래스별 콘텐츠 표현자(c_k^x, c_k^y)의 클래스를 식별하여 메모리(500)에 저장된 다수의 아이템 중 대응하는 클래스에 할당된 M_k 개의 아이템 각각에 포함된 아이템 키(k)를 리드한다. 업데이트 유사도 계산부(611)는 K개의 클래스별 콘텐츠 표현자(c_k^x, c_k^y) 각각에서 픽셀 단위의 픽셀 콘텐츠 표현자(c_p^x, c_p^y)를 추출한다.

[0079] 그리고 각 픽셀 콘텐츠 표현자(c_p^x, c_p^y)와 해당 클래스에 할당된 아이템 키(k_n) 각각 사이의 유사도($d(c_p^x, k_n), d(c_p^y, k_n)$)를 수학적 식 4에 따라 계산한다.

수학적 식 4

$$d(c_p^x, k_n) = \frac{c_p^x k_n^T}{\|c_p^x\|_2 \|k_n\|_2}$$

$$d(c_p^y, k_n) = \frac{c_p^y k_n^T}{\|c_p^y\|_2 \|k_n\|_2}$$

[0080]

[0081] 업데이트 가중치 계산부(612)는 업데이트 유사도 계산부(611)에서 계산된 유사도($d(c_p^x, k_n), d(c_p^y, k_n)$)를 기반으로 각 도메인의 해당 클래스에서 M_k 개의 아이템 키(k_n) 각각에 대한 업데이트 가중치($\beta_{p,n}^x, \beta_{p,n}^y$)를 수학적 식 5에 따라 계산한다.

수학식 5

$$\beta_{p,n}^x = \frac{\exp(d(c_p^x, k_n))}{\sum_{p'=1}^{N_k} \exp(d(c_{p'}^x, k_n))}$$

$$\beta_{p,n}^y = \frac{\exp(d(c_p^y, k_n))}{\sum_{p'=1}^{N_{k'}} \exp(d(c_{p'}^y, k_n))}$$

[0082]

[0083]

각 도메인에서 클래스 내의 M_k 개의 아이템 $k_i(k_n)$ 각각에 대한 M_k 개의 업데이트 가중치($\beta_{p,n}^x, \beta_{p,n}^y$)가 계산되면, 업데이트 값 계산부(613)는 메모리(500)에서 M_k 개의 아이템 $k_i(k_n)$ 를 수학식 6에 따라 업데이트하여 업데이트 키(\hat{k}_n)를 획득한다.

수학식 6

$$\hat{k}_n = \left\| k_n + \sum_{p'=1}^{N_k} \beta_{p',n}^x c_{p'}^x + \sum_{p'=1}^{N_{k'}} \beta_{p',n}^y c_{p'}^y \right\|_2$$

[0084]

[0085]

그리고 각 아이템 $k_i(k_n)$ 에 매칭된 2개의 스타일 값(v_n^x, v_n^y)을 리드하고, 인코딩부(200)에서 클래스별 스타일 표현자(s_k^x, s_k^y)를 인가받는다. 이후 수학식 7과 같이, 클래스별 스타일 표현자(s_k^x, s_k^y) 각각에서 픽셀 단위로 추출된 픽셀 타겟 스타일 표현자(s_p^x, s_p^y)에 대응하는 업데이트 가중치($\beta_{p,n}^x, \beta_{p,n}^y$)를 가중합하여, 리드된 2개의 스타일 값(v_n^x, v_n^y)에 가산함으로써, 업데이트 스타일 값(\hat{v}_n^x, \hat{v}_n^y)을 획득한다.

수학식 7

$$\hat{v}_n^x = \left\| v_n^x + \sum_{p'=1}^{N_k} \beta_{p',n}^x s_{p'}^x \right\|_2$$

$$\hat{v}_n^y = \left\| v_n^y + \sum_{p'=1}^{N_{k'}} \beta_{p',n}^y s_{p'}^y \right\|_2$$

[0086]

[0087]

업데이트 키(\hat{k}_n^x)와 업데이트 스타일 값(\hat{v}_n^x, \hat{v}_n^y)이 획득되어 아이템이 업데이트되면, 업데이트된 아이템을 메모리(500)에 다시 저장한다.

[0088]

즉 업데이트부(600)는 서로 다른 도메인의 입력 영상(I^x, I^y)이 입력되어 서로 다른 도메인에서의 객체의 구조적 특징인 클래스별 콘텐츠 표현자(c_k^x, c_k^y)가 획득되면, 각 도메인에서 획득된 콘텐츠 표현자(c_k^x, c_k^y)와 기존에 메모리에 저장된 대응하는 클래스의 대표 구조적 특징인 아이템 $k_i(k_n)$ 들 사이의 유사도에 따른 가중치($\beta_{p,n}^x, \beta_{p,n}^y$)를 계산하여, 현재 획득된 클래스별 콘텐츠 표현자(c_k^x, c_k^y)에 각 픽셀별로 가중하여 기존 아이템 $k_i(k_n)$

에 추가로 반영함으로써 업데이트 아이템 키(\hat{k}_n^x)를 획득한다. 유사하게 가중치($\beta_{p,n}^x, \beta_{p,n}^y$)를 현재 획득된 클래스별 스타일 표현자(s_k^x, s_k^y)에 각 픽셀별로 가중하여 기존 스타일 값(v_n^x, v_n^y) 각각에 추가로 반영함으로써 업데이트 스타일 값(\hat{v}_n^x, \hat{v}_n^y)을 획득한다.

[0089] 상기한 업데이트부(600)를 이용하여, 메모리(500)에 저장된 다수의 아이템은 업데이트할 수 있으나, 인공 신경망은 학습되지 않았다. 또한 메모리(500)에 저장되는 다수의 아이템이 각각 해당 클래스에서 서로 상이한 구조적 특징 및 표현적 특징을 가져 각 특징을 대표하도록, 인코딩부(200)는 클래스별 콘텐츠 표현자(c_k^x, c_k^y)와 클래스별 스타일 표현자(s_k^x, s_k^y)를 추출할 수 있어야 한다.

[0090] 이에 본 실시예의 영상 변환 장치는 학습을 위한 학습부(미도시)를 더 포함할 수 있다. 학습부는 영상 변환 장치의 학습이 종료되면 제거될 수 있다.

[0091] 학습부는 인코딩부(200)에서 추출된 클래스별 콘텐츠 표현자(c_k^x, c_k^y)의 N개의 픽셀 콘텐츠 표현자(c_p^x, c_p^y)와 클래스별 스타일 표현자(s_k^x, s_k^y)의 N개의 픽셀 타겟 스타일 표현자(s_p^x, s_p^y)가 각각 메모리(500)에 저장된 M개의 아이템 중 하나의 아이템에 대한 아이템 키(k_{p+}) 및 스타일 값(v_{p+}^x, v_{p+}^y)과는 차이가 작아지는 반면, 나머지 아이템에 대한 아이템 키(k_n) 및 스타일 값(v_n^x, v_n^y)과는 차이가 증가되도록 하는 키 손실(L_k) 및 스타일 손실(L_v)을 각각 수학적 식 8 및 9에 따라 계산한다.

수학적 식 8

$$\mathcal{L}_k = - \sum_{p=1}^N \log \frac{\exp(c_p^x \cdot k_{p+}/\tau)}{\sum_{n=1}^{M-1} \exp(c_p^x \cdot k_n/\tau)} - \sum_{p=1}^N \log \frac{\exp(c_p^y \cdot k_{p+}/\tau)}{\sum_{n=1}^{M-1} \exp(c_p^y \cdot k_n/\tau)}$$

수학적 식 9

$$\mathcal{L}_v = - \sum_{p=1}^N \log \frac{\exp(s_p^x \cdot v_{p+}^x/\tau)}{\sum_{n=1}^{M-1} \exp(s_p^x \cdot v_n^x/\tau)} - \sum_{p=1}^N \log \frac{\exp(s_p^y \cdot v_{p+}^y/\tau)}{\sum_{n=1}^{M-1} \exp(s_p^y \cdot v_n^y/\tau)}$$

[0094] 여기서 k_{p+} 와 v_{p+}^x 및 v_{p+}^y 는 메모리에 저장된 M개의 아이템 각각의 아이템 키(k_n)와 스타일 값(v_n^x, v_n^y) 중 콘텐츠 표현자(c_p^x, c_p^y)와 픽셀 타겟 스타일 표현자(s_p^x, s_p^y)에 가장 유사한 아이템 키와 스타일 값을 나타내는 포지티브 샘플을 나타낸다.

[0095] 또한 학습부는 타겟 스타일 표현자(\hat{s}^y, \hat{s}^x)가 정상적으로 획득되고, 출력 영상 생성부(400)가 정상적으로 요구되는 타겟 도메인 스타일의 출력 영상(\hat{I}^y, \hat{I}^x)을 생성하는지 여부를 판별하기 위해, 자기 재구성 손실(L^{self})과 순환 재구성 손실(L^{cyc})을 수학적 식 10 및 11에 따라 계산할 수 있다.

수학적 식 10

$$\mathcal{L}^{\text{self}} = \|G^x(c^x, \hat{s}^x) - I^x\|_1 + \|G^y(c^y, \hat{s}^y) - I^y\|_1$$

수학식 11

$$\mathcal{L}^{cyc} = \|G^x(\hat{c}^y, \hat{s}^x) - I^x\|_1 + \|G^y(\hat{c}^x, \hat{s}^y) - I^y\|_1$$

[0097]

[0098]

여기서 $G^x()$, $G^y()$ 는 각각 제1 출력 영상 생성부(410)와 제2 출력 영상 생성부(420)의 신경망 연산을 표현한 함수이다.

[0099]

자기 재구성 손실(L^{self})은 수학식 10에 나타난 바와 같이, 제1 출력 영상 생성부(410)와 제2 출력 영상 생성부(420)에 동일한 도메인에 대한 콘텐츠 표현자(c^x , c^y)와 타겟 스타일 표현자(\hat{s}^x , \hat{s}^y)가 입력된 경우, 제1 출력 영상 생성부(410)와 제2 출력 영상 생성부(420)에서 출력되는 자기 재구성 영상($G^x(c^x, \hat{s}^x)$, $G^y(c^y, \hat{s}^y)$)은 입력 영상(I^x , I^y)과 동일해야 한다는 것을 나타낸다.

[0100]

그리고 순환 재구성 손실(L^{cyc})은 입력 영상(I^x , I^y)을 타겟 도메인으로 변환된 영상(\hat{I}^y , \hat{I}^x)을 입력 영상으로 순환하여 입력하는 경우, 제1 출력 영상 생성부(410)와 제2 출력 영상 생성부(420)의 출력($G^x(\hat{c}^y, \hat{s}^x)$, $G^y(\hat{c}^x, \hat{s}^y)$)은 입력 영상(I^x , I^y)과 동일해야 한다는 것을 나타낸다.

[0101]

학습부는 키 손실(L_k)과 스타일 손실(L_v), 자기 재구성 손실(L^{self}) 및 순환 재구성 손실(L^{cyc})이 계산되면, 계산된 키 손실(L_k)과 스타일 손실(L_v), 자기 재구성 손실(L^{self}) 및 순환 재구성 손실(L^{cyc})을 기지정된 방식으로 가중합하여 총 손실(L)을 계산하고, 계산된 총 손실을 역전파하여 본 실시예의 영상 변환 장치에 포함된 인공 신경망에 대한 학습을 수행할 수 있다.

[0102]

여기서는 설명의 편의를 위하여 업데이트부(600)와 학습부를 구분하여 설명하였으나, 업데이트부(600) 또한 학습부에 포함될 수도 있다.

[0103]

그리고 본 실시예의 영상 변환 장치에서 학습은 학습 기간 동안 계속하여 인가되는 입력 영상(I^x , I^y) 각각에 대해 우선 업데이트부(600)가 메모리에 대한 업데이트를 우선 수행하고, 이후, 학습부가 손실을 역전파하는 방식으로 수행될 수 있다.

[0104]

도 9는 본 실시예의 영상 변환 장치와 기존의 영상 변환 장치의 성능을 비교한 결과를 나타낸다.

[0105]

도 9에서는 맑은 날씨에서 획득된 입력 영상에 대해 비오는 날씨와 흐린 날씨의 영상으로 변환한 결과를 도시하였다. 도 9에 도시된 바와 같이, 기존의 방식에 따른 영상 변환 장치의 경우, 도메인의 변화로 인해 객체인 차량의 형상이 정상적으로 출력되지 않은 반면, 본 실시예에 따른 영상 변환 장치는 각 객체의 특성을 고려하여 개별적인 스타일을 적용하여 변환함에 따라 변환된 영상에서도 객체가 매우 선명하게 획득되었음을 알 수 있다.

[0106]

도 10은 본 발명의 일 실시예에 따른 영상 변환 방법을 나타낸다.

[0107]

도 2 내지 도 8을 참조하면, 도 10에 도시된 본 발명의 영상 변환 방법은 우선 타겟 도메인의 영상으로 변환되어야 하는 소스 도메인의 입력 영상(I^x)을 획득한다(S11).

[0108]

그리고 미리 학습된 인공 신경망을 이용하여 획득된 입력 영상(I^x)에 대해 신경망 연산을 수행하여, 각 객체의 클래스에 따른 구조적 특징을 나타내는 다수의 클래스별 콘텐츠 표현자(c_k^x)를 획득한다(S12). 여기서 클래스별 콘텐츠 표현자(c_k^x)는 입력 영상(I^x , I^y)의 전체적인 구조적 특징을 나타내는 콘텐츠 표현자(c^x)를 획득한 후, 콘텐츠 표현자(c^x)에 포함된 각 객체의 객체 영역을 객체의 클래스별로 구분하여 획득될 수 있다.

[0109]

다수의 클래스별 콘텐츠 표현자(c_k^x)가 획득되면, 다수의 클래스별 콘텐츠 표현자(c_k^x) 각각의 클래스에 따라 메모리(500)에 다수의 클래스 각각에 대해 적어도 하나씩 할당되어 저장된 아이템 중 해당 클래스의 모든 아이템

을 리드한다(S13). 여기서 각 아이템에는 해당 클래스의 객체의 대표적인 구조적 특징을 나타내는 벡터인 아이템 키(k)와 타겟 도메인에서 객체의 대표적인 스타일 특징을 나타내는 벡터인 스타일 값(v^y)이 매칭되어 포함된다. 경우에 따라서는 아이템에 소스 도메인에서의 객체의 대표적인 스타일 특징을 나타내는 벡터인 스타일 값(v^x)도 함께 매칭되어 포함될 수 있다.

[0110] 메모리(500)에서 해당 클래스의 모든 아이템이 리드되면, 각 클래스별 콘텐츠 표현자(c_k^x)에서 픽셀 단위의 픽셀 콘텐츠 표현자(c_p^x)를 추출하고, 픽셀 콘텐츠 표현자(c_p^x)와 리드된 아이템 각각의 아이템 키(k_n) 사이의 유사도($d(c_p^x, k_n)$)를 수학적 식 1과 같이 계산한다(S14). 그리고 계산된 픽셀 콘텐츠 표현자(c_p^x)와 아이템 키(k_n) 사이의 유사도($d(c_p^x, k_n)$)를 기반으로 수학적 식 2에 따라 아이템 키(k_n) 각각에 대한 리드 가중치($\alpha_{p,n}^x$)를 계산한다(S15).

[0111] 리드 가중치($\alpha_{p,n}^x$)가 계산되면, 적어도 하나의 아이템의 각 아이템 키(k_n)에 매칭된 적어도 하나의 스타일 값(v_n^y)에 대응하는 리드 가중치($\alpha_{p,n}^x$)를 수학적 식 3과 같이 가중합하여, 픽셀별 타겟 스타일을 나타내는 픽셀 타겟 스타일 표현자(\hat{s}_p^y)를 획득한다(S16).

[0112] 다수의 클래스 각각에 대한 픽셀 단위의 픽셀 타겟 스타일 표현자(\hat{s}_p^y)가 획득되면, 모든 클래스의 픽셀 타겟 스타일 표현자(\hat{s}_p^y)를 해당 픽셀 위치에 배치하여 타겟 스타일 표현자(\hat{s}^y)를 획득하고, 이후 입력 영상(I^x)의 구조적 특징을 나타내는 콘텐츠 표현자(c^x)와 각 타겟 스타일 표현자(\hat{s}^y)를 결합하여 결합 타겟 스타일 표현자(c^x, \hat{s}^y)를 획득한다. 즉 소스 도메인의 입력 영상(I^x)의 구조적 특징과 타겟 도메인의 스타일 특징을 결합한다.

[0113] 그리고 미리 학습된 인공 신경망에 결합 타겟 스타일 표현자(c^x, \hat{s}^y)를 입력하여 신경망 연산을 수행함으로써, 소스 도메인의 입력 영상(I^x)이 타겟 도메인의 영상으로 재구성된 출력 영상(\hat{I}^y)를 생성한다.

[0114] 다만 도 10의 영상 변환 방법을 수행하기 위해서는 메모리(500)에 저장된 다수의 아이템 각각의 아이템 키(k)가 대응하는 클래스의 객체의 구조적 특징을 대표하고, 스타일 값(v^y, v^x)는 해당 도메인에서 객체의 스타일 특징을 대표할 수 있도록 미리 업데이트되어야 한다. 또한 인공 신경망이 미리 학습되어야 한다.

[0115] 이에 도 10의 영상 변환 방법을 수행하기 이전 학습 단계를 우선 수행하여 메모리(500)에 저장되는 아이템을 업데이트하고, 인공 신경망을 학습시킬 수 있다.

[0116] 도 11은 도 10의 영상 변환 방법을 위한 학습 단계의 일 예를 나타낸다.

[0117] 도 11을 참조하면, 학습 단계에서는 우선 서로 다른 도메인의 2개의 입력 영상(I^x, I^y) 중 적어도 하나를 획득한다(S21). 그리고 미리 학습된 인공 신경망을 이용하여 획득된 입력 영상(I^x, I^y)에 대해 신경망 연산을 수행하여, 각 객체의 클래스에 따른 구조적 특징을 나타내는 다수의 클래스별 콘텐츠 표현자(c_k^x, c_k^y)와 함께 클래스에 따른 스타일 특징을 나타내는 다수의 클래스별 스타일 표현자(s_k^x, s_k^y)를 획득한다(S22).

[0118] 그리고 메모리(500)에 저장된 다수의 아이템 중 획득된 다수의 클래스별 콘텐츠 표현자(c_k^x, c_k^y)와 다수의 클래스별 스타일 표현자(s_k^x, s_k^y) 각각에 대응하는 클래스의 모든 아이템을 리드하고, 각 클래스별 콘텐츠 표현자(c_k^x, c_k^y)의 픽셀 콘텐츠 표현자(c_p^x, c_p^y)와 리드된 아이템 각각의 아이템 키(k_n) 사이의 유사도($d(c_p^x, k_n)$, $d(c_p^y, k_n)$)를 계산한다(S23).

- [0119] 유사도($d(c_p^x, k_n), d(c_p^y, k_n)$)가 계산되면, 계산된 유사도를 이용하여 리드 가중치($a_{p,n}^x, a_{p,n}^y$)를 계산하고, 계산된 리드 가중치($a_{p,n}^x, a_{p,n}^y$)에 따라 타겟 스타일 표현자($\hat{\mathbf{s}}^y, \hat{\mathbf{s}}^x$)를 획득하여 입력 영상(I^x, I^y)과 상이한 도메인의 출력 영상($\hat{\mathbf{I}}^y, \hat{\mathbf{I}}^x$)을 생성한다(S24).
- [0120] 한편, 계산된 유사도($d(c_p^x, k_n), d(c_p^y, k_n)$)를 이용하여 아이템 키(k_n)를 업데이트한 업데이트 가중치($\beta_{p,n}^x, \beta_{p,n}^y$)를 계산한다(S25).
- [0121] 업데이트 가중치($\beta_{p,n}^x, \beta_{p,n}^y$)가 계산되면, 업데이트 가중치($\beta_{p,n}^x, \beta_{p,n}^y$)와 픽셀 콘텐츠 표현자(c_p^x, c_p^y)를 이용하여 수학적 식 6에 따라 업데이트 아이템 키(\hat{k}_n)를 계산하여 획득한다(S26).
- [0122] 그리고 업데이트 가중치($\beta_{p,n}^x, \beta_{p,n}^y$)와 클래스별 스타일 표현자(s_k^x, s_k^y) 각각에서 픽셀 단위로 추출된 픽셀 타겟 스타일 표현자(s_p^x, s_p^y)를 이용하여 각 도메인에 따른 스타일 값(v^y, v^x)을 업데이트한 업데이트 스타일 값(\hat{v}_n^x, \hat{v}_n^y)을 획득한다(S27).
- [0123] 업데이트된 아이템 키(\hat{k}_n)와 대응하는 스타일 값(\hat{v}_n^x, \hat{v}_n^y)을 매칭하여 획득되는 아이템을 메모리(500)에 저장한다(S28). 즉 기존에 저장된 아이템의 아이템 키(k)와 스타일 값(v^y, v^x)을 업데이트된 아이템 키(\hat{k}_n)와 대응하는 스타일 값(\hat{v}_n^x, \hat{v}_n^y)으로 대체하여 저장한다.
- [0124] 그리고 픽셀 콘텐츠 표현자(c_p^x, c_p^y)와 픽셀 타겟 스타일 표현자(s_p^x, s_p^y)가 각각 메모리(500)에 저장된 아이템 중 하나의 아이템에 대한 아이템 키(k_{p+}) 및 스타일 값(v_{p+}^x, v_{p+}^y)과는 차이가 작아지는 반면, 나머지 아이템에 대한 아이템 키(k_n) 및 스타일 값(v_n^x, v_n^y)과는 차이가 증가되도록 하는 키 손실(L_k) 및 스타일 손실(L_v)을 각각 수학적 식 8 및 9에 따라 계산한다(S29).
- [0125] 또한 결합 타겟 스타일 표현자($(c^x, \hat{\mathbf{s}}^y), (c^y, \hat{\mathbf{s}}^x)$)를 인가받아 타겟 도메인에 대응하는 스타일의 출력 영상($\hat{\mathbf{I}}^y, \hat{\mathbf{I}}^x$)을 생성하는 인공 신경망에 동일한 도메인에 대한 콘텐츠 표현자(c^x, c^y)와 타겟 스타일 표현자($\hat{\mathbf{s}}^x, \hat{\mathbf{s}}^y$)가 결합된 결합 타겟 스타일 표현자($(c^x, \hat{\mathbf{s}}^x), (c^y, \hat{\mathbf{s}}^y)$)를 인가하여 출력되는 자기 재구성 영상($G^x(c^x, \hat{\mathbf{s}}^x), G^y(c^y, \hat{\mathbf{s}}^y)$)과 입력 영상(I^x, I^y) 사이의 차이로 자기 재구성 손실(L^{self})을 계산하고, 입력 영상(I^x, I^y)에서 타겟 도메인으로 변환된 영상($\hat{\mathbf{I}}^y, \hat{\mathbf{I}}^x$)을 다시 소스 도메인으로 변환하여 획득되는 순환 재구성 영상($G^x(\hat{c}^y, \hat{\mathbf{s}}^x), G^y(\hat{c}^x, \hat{\mathbf{s}}^y)$)과 입력 영상(I^x, I^y) 사이의 차이로 순환 재구성 손실(L^{cyc})을 계산한다(S30).
- [0126] 그리고 키 손실(L_k)과 스타일 손실(L_v), 자기 재구성 손실(L^{self}) 및 순환 재구성 손실(L^{cyc})을 기지정된 방식으로 가중합하여 총 손실(L)을 계산하고, 계산된 총 손실을 역전파하여 학습을 수행한다(S31).
- [0127] 여기서 학습은 기지정된 횟수 또는 총 손실이 기지정된 기준 손실 이하가 될때까지 반복하여 수행될 수 있다.
- [0128] 본 발명에 따른 방법은 컴퓨터에서 실행시키기 위한 매체에 저장된 컴퓨터 프로그램으로 구현될 수 있다. 여기서 컴퓨터 판독가능 매체는 컴퓨터에 의해 액세스 될 수 있는 임의의 가용 매체일 수 있고, 또한 컴퓨터 저장 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함하며, ROM(판독 전용 메모리), RAM(랜덤 액세스 메모리), CD(컴팩트 디스크)-ROM, DVD(디

지털 비디오 디스크)-ROM, 자기 테이프, 플로피 디스크, 광데이터 저장장치 등을 포함할 수 있다.

[0129] 본 발명은 도면에 도시된 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다.

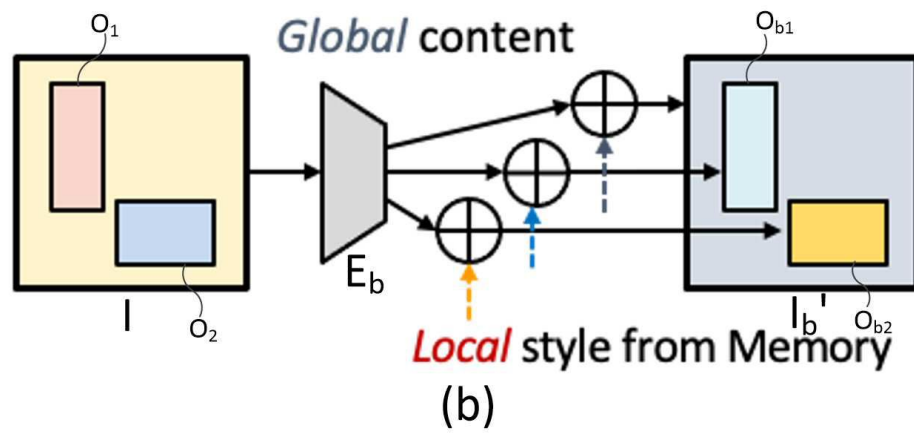
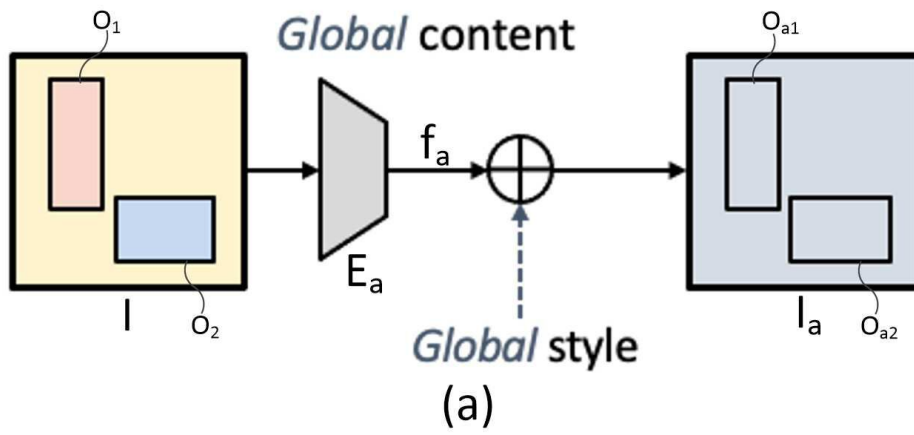
[0130] 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 청구범위의 기술적 사상에 의해 정해져야 할 것이다.

부호의 설명

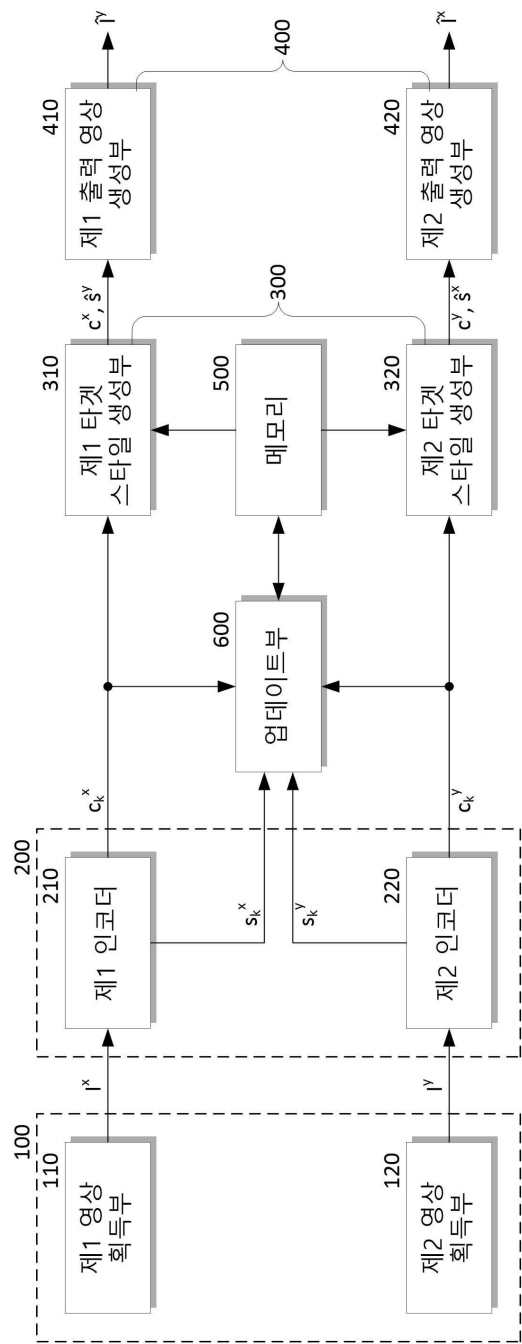
[0131]	100: 영상 획득부	110: 제1 영상 획득부
	120: 제2 영상 획득부	200: 인코딩부
	210: 제1 인코더	211: 객체 식별부
	212: 콘텐츠 표현자 획득부	213: 콘텐츠 클러스터링부
	214: 스타일 표현자 획득부	215: 스타일 클러스터링부
	220: 제2 인코더	300: 타겟 스타일 생성부
	310: 제1 타겟 스타일 생성부	311: 리드 유사도 계산부
	312: 리드 가중치 계산부	313: 타겟 스타일 표현자 획득부
	314: 타겟 스타일 결합부	320: 제2 타겟 스타일 생성부
	400: 출력 영상 생성부	500: 메모리
	600: 업데이트부	610: 업데이트 유사도 계산부
	620: 업데이트 가중치 계산부	630: 업데이트 값 계산부

도면

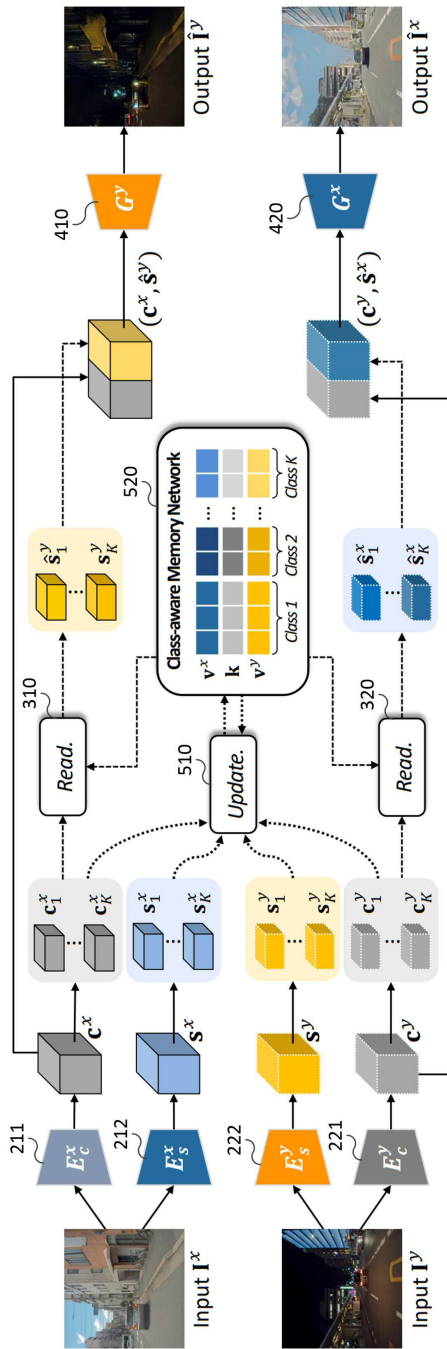
도면1



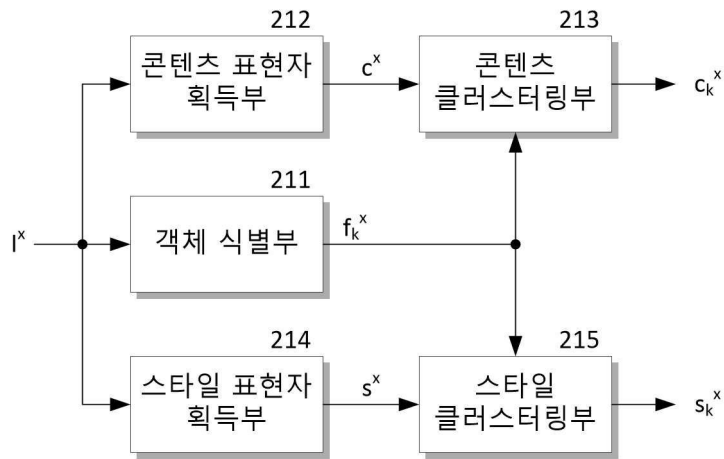
도면2



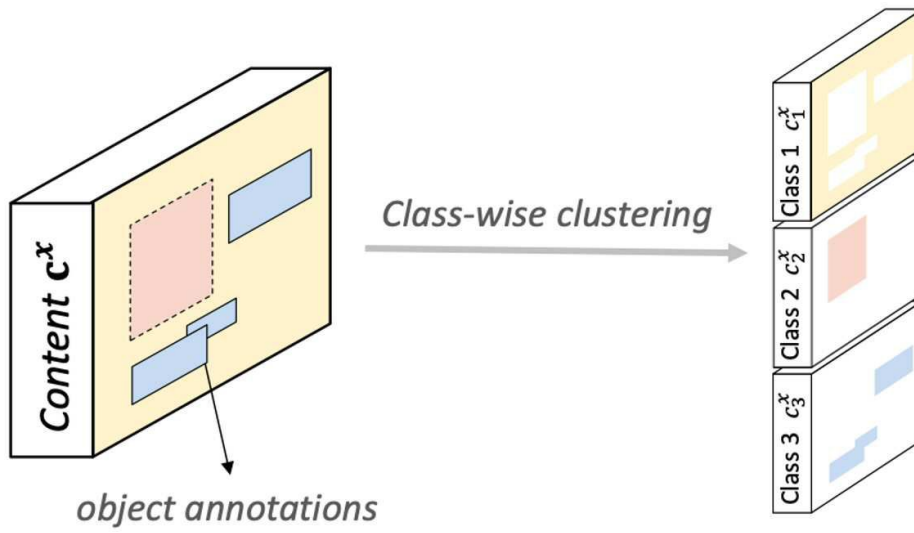
도면3



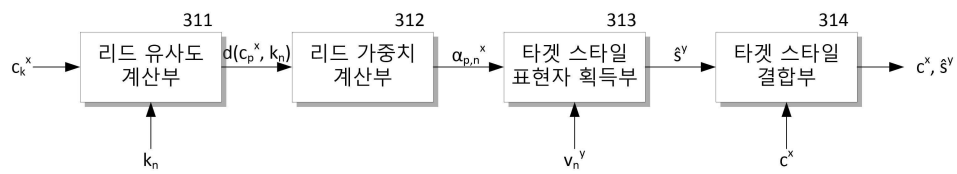
도면4



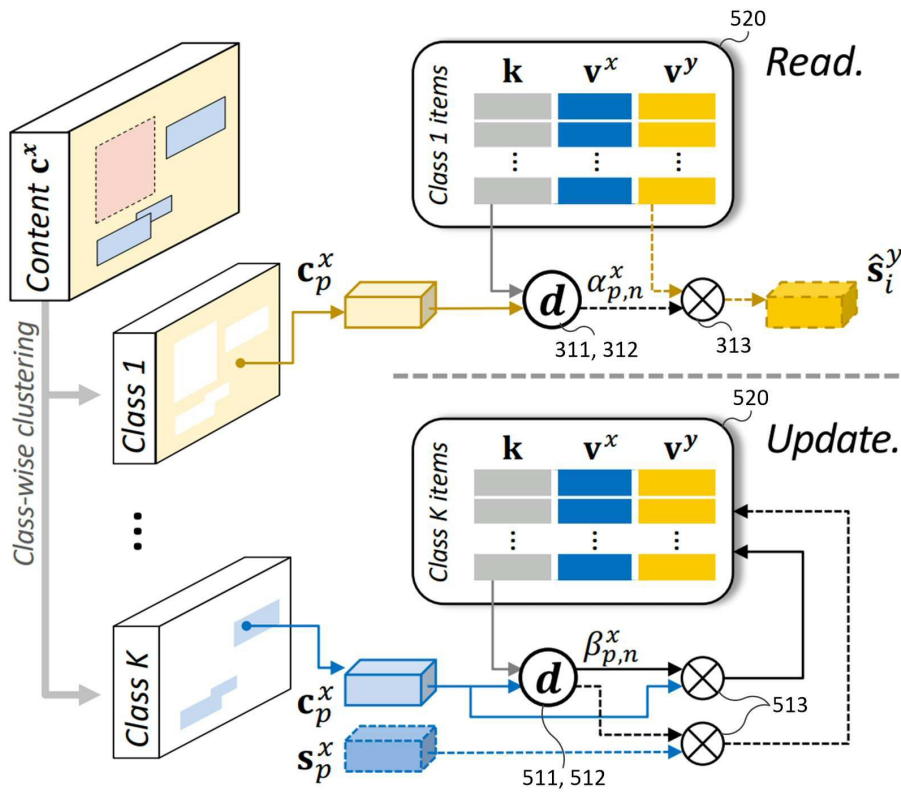
도면5



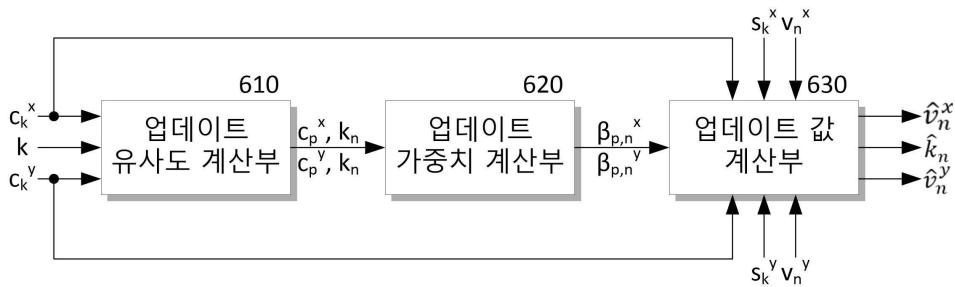
도면6



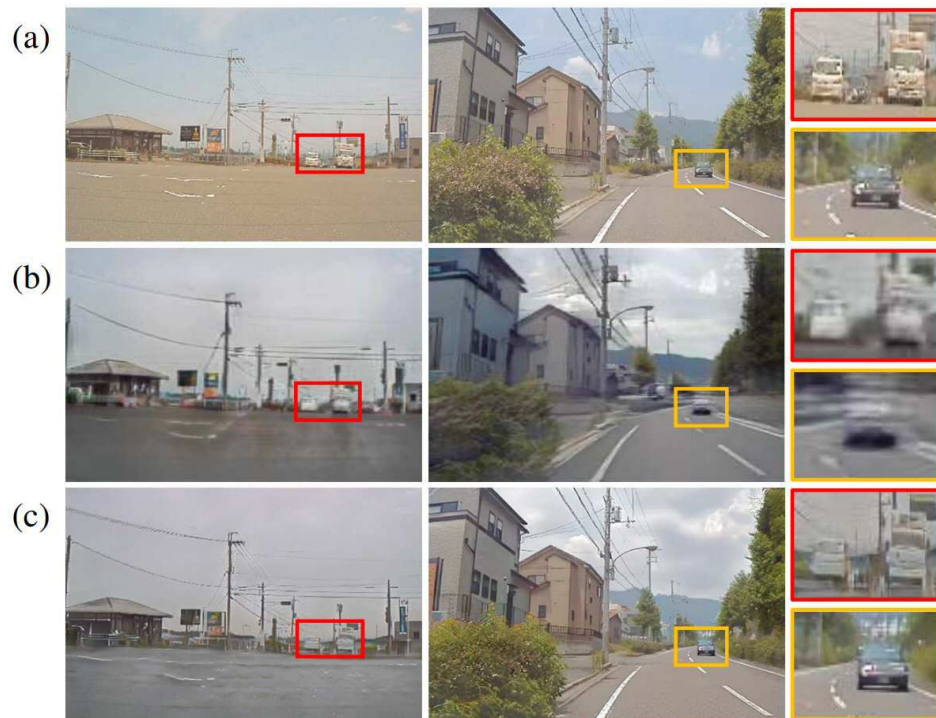
도면7



도면8



도면9



도면10



도면11

