(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0123218
(43) 공개일자 2024년08월13일

(51) 국제특허분류(Int. Cl.)
G06N 3/096 (2023.01) *G06N 3/0455* (2023.01)
G06N 3/0895 (2023.01) *G06T 5/00* (2024.01)
(52) CPC특허분류
G06N 3/096 (2023.01)
G06N 3/0455 (2023.01)
(21) 출원번호 10-2023-0089237
(22) 출원일자 2023년07월10일
심사청구일자 2023년07월10일
(30) 우선권주장
1020230015695 2023년02월06일 대한민국(KR)

(71) 출원인
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
민동보
서울특별시 서대문구 북아현로1길 17, 203동 706호(북아현동,이편한세상신촌)
최혜송
서울특별시 서대문구 신촌로 163, 1107호(대현동)
(뒷면에 계속)
(74) 대리인
특허법인 무한

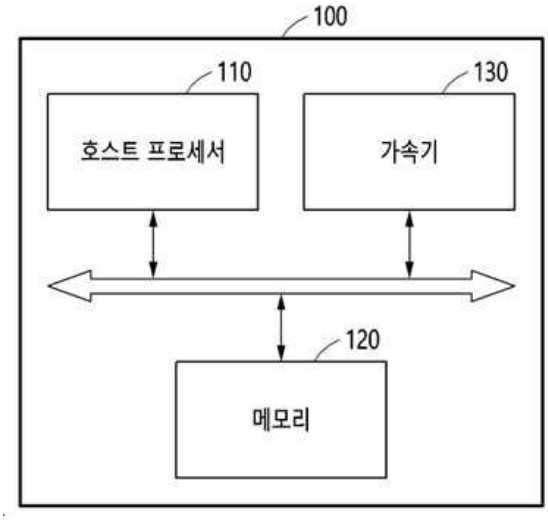
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **마스킹된 이미지를 복원하는 사전 훈련 모델을 생성하는 전자 장치 및 그 동작 방법**

(57) 요 약

마스킹된 이미지를 복원하는 사전 훈련 모델을 생성하는 전자 장치 및 그 동작 방법이 개시된다. 본 발명의 일 실시예에 따른 전자 장치의 동작 방법은, 입력 이미지를 기반으로 생성된 소스 뷰(source view) 및 타겟 뷰(target view) 각각을 소스 인코더(source encoder) 및 타겟 인코더(target encoder) 각각에 입력하는 단계, 차별 학습(discrimination learning) 및 상기 입력 이미지에 기반하여 상기 소스 인코더의 출력 및 상기 타겟 인코더의 출력으로부터 손실을 계산하는 단계, 및 상기 손실이 적용된 상기 소스 인코더를 포함하는 사전 훈련 모델(pre-trained model)을 생성하는 단계를 포함할 수 있다.

대 표 도 - 도1



(52) CPC특허분류
G06N 3/0895 (2023.01)
G06T 5/00 (2024.01)
(72) 발명자
이헌상
서울특별시 서대문구 연대동문길 111(대신동)
박혜진
서울특별시 동대문구 한천로2길 27-9, 302호(장안동, 리치빌)

명 세 서

청구범위

청구항 1

전자 장치의 동작 방법에 있어서,

입력 이미지를 기반으로 생성된 소스 뷰(source view) 및 타겟 뷰(target view) 각각을 소스 인코더(source encoder) 및 타겟 인코더(target encoder) 각각에 입력하는 단계;

차별 학습(discrimination learning) 및 상기 입력 이미지에 기반하여 상기 소스 인코더의 출력 및 상기 타겟 인코더의 출력으로부터 손실을 계산하는 단계; 및

상기 손실이 적용된 상기 소스 인코더를 포함하는 사전 훈련 모델(pre-trained model)을 생성하는 단계

를 포함하는, 동작 방법.

청구항 2

제1항에 있어서,

상기 입력 이미지를 기반으로 상기 소스 뷰를 생성하는 단계; 및

상기 입력 이미지를 기반으로 상기 타겟 뷰를 생성하는 단계를 더 포함하고,

상기 소스 뷰는,

상기 입력 이미지가 분할된 복수의 소스 패치들(source patches) 및 상기 복수의 소스 패치들 간의 관계 정보를 포함하는 소스 분류 토큰(source classification token)을 포함하고,

상기 복수의 소스 패치들은,

복수의 소스 이미지 토큰들 및 복수의 소스 마스크 토큰들을 포함하고,

상기 타겟 뷰는,

상기 입력 이미지가 분할된 복수의 타겟 패치들(target patches) 및 상기 복수의 타겟 패치들 간의 관계 정보를 포함하는 타겟 분류 토큰(target classification token)을 포함하고,

상기 복수의 타겟 패치들은,

복수의 타겟 이미지 토큰들 및 복수의 타겟 마스크 토큰들을 포함하는,

동작 방법.

청구항 3

제2항에 있어서,

상기 입력 이미지를 기반으로 상기 소스 뷰를 생성하는 단계는,

토큰화(tokenize)를 통해 상기 입력 이미지를 복수의 패치들로 분할하는 단계;

상기 복수의 패치들 중 일부 패치들에 대해 랜덤 마스킹을 수행하여, 상기 랜덤 마스킹된 상기 복수의 소스 마스크 토큰들 및 상기 랜덤 마스킹되지 않은 상기 복수의 소스 이미지 토큰들을 포함하는 상기 복수의 소스 패치들을 생성하는 단계;

상기 복수의 소스 패치들 간의 관계 정보를 포함하는 상기 소스 분류 토큰을 생성하는 단계; 및

상기 소스 분류 토큰 및 상기 복수의 소스 패치들을 포함하는 상기 소스 뷰를 생성하는 단계

를 포함하는, 동작 방법.

청구항 4

제2항에 있어서,

상기 입력 이미지를 기반으로 상기 타겟 뷰를 생성하는 단계는,

토큰화를 통해 상기 입력 이미지를 복수의 패치들로 분할하는 단계;

상기 복수의 패치들에 대해 상기 소스 뷰의 랜덤 마스킹과 겹치지 않도록 마스킹을 수행하여, 상기 마스킹된 상기 복수의 타겟 마스크 토큰들 및 상기 마스킹되지 않은 상기 복수의 타겟 이미지 토큰들을 포함하는 상기 복수의 타겟 패치들을 생성하는 단계;

상기 복수의 타겟 패치들 간의 관계 정보를 나타내는 상기 타겟 분류 토큰을 생성하는 단계; 및

상기 타겟 분류 토큰 및 상기 복수의 타겟 패치들을 포함하는 상기 타겟 뷰를 생성하는 단계

를 포함하는, 동작 방법.

청구항 5

제2항에 있어서,

복수의 소스 출력 패치들을 디코더에 입력하여 상기 복수의 소스 마스크 토큰들을 복원하는 단계; 및

상기 복원된 상기 복수의 소스 마스크 토큰들 각각의 불확실성을 계산하는 단계를 더 포함하고,

상기 복수의 소스 출력 패치들은,

상기 소스 인코더의 상기 출력 중 상기 복수의 소스 패치들에 대한 출력인,

동작 방법.

청구항 6

제5항에 있어서,

상기 복원된 상기 복수의 소스 마스크 토큰들 각각의 상기 불확실성을 계산하는 단계는,

상기 복원된 상기 복수의 소스 마스크 토큰들 각각과 상기 복수의 소스 마스크 토큰들의 위치에 대응하는 상기 입력 이미지의 패치들 간의 거리인 불확실성을 계산하는 단계

를 포함하는, 동작 방법.

청구항 7

제5항에 있어서,

상기 소스 인코더의 출력 및 상기 타겟 인코더의 출력으로부터 손실을 계산하는 단계는,

상기 복수의 소스 출력 패치들 및 복수의 타겟 출력 패치들 중 상기 불확실성이 임계값 미만인 패치들에 대응하는 토큰들을 로컬 차별 학습 모듈에 입력하여 로컬 손실을 계산하는 단계;

소스 출력 분류 토큰 및 타겟 출력 분류 토큰을 글로벌 차별 학습 모듈에 입력하여 글로벌 손실을 계산하는 단계; 및

상기 복원된 상기 복수의 소스 마스크 토큰들과 상기 입력 이미지를 이용하여 회귀 손실을 계산하는 단계

를 포함하고,

상기 복수의 타겟 출력 패치들은,

상기 타겟 인코더의 상기 출력 중 상기 복수의 타겟 패치들에 대응하는 출력이고,

상기 소스 출력 분류 토큰은,

상기 소스 인코더의 출력 중 상기 소스 분류 토큰에 대응하는 출력이고,

상기 타겟 출력 분류 토큰은,

상기 타겟 인코더의 출력 중 상기 타겟 분류 토큰에 대응하는 출력인,

동작 방법.

청구항 8

제7항에 있어서,

상기 손실이 적용된 상기 소스 인코더를 포함하는 상기 사전 훈련 모델을 생성하는 단계는,

상기 로컬 손실, 상기 글로벌 손실 및 상기 회귀 손실을 상기 소스 인코더에 반영하여 상기 사전 훈련 모델을 생성하는, 동작 방법.

청구항 9

제1항에 있어서,

상기 사전 훈련 모델에 포함되는 상기 손실이 적용된 상기 소스 인코더의 파라미터를 초기 파라미터로 하여 타겟 모델(target model)을 트레이닝하는 단계를 더 포함하는, 동작 방법.

청구항 10

제1항에 있어서,

상기 타겟 인코더의 파라미터들이 상기 소스 인코더의 파라미터를 이용한 지수 이동 평균(exponential moving average)을 통해 업데이트되는 단계를 더 포함하는, 동작 방법.

청구항 11

전자 장치의 동작 방법에 있어서,

입력 이미지를 기반으로 생성된 소스 뷰 및 타겟 뷰 각각을 소스 인코더 및 타겟 인코더 각각에 입력하는 단계;

차별 학습 및 상기 입력 이미지에 기반하여 상기 소스 인코더의 출력 및 상기 타겟 인코더의 출력으로부터 손실을 계산하는 단계;

상기 손실이 적용된 상기 소스 인코더를 포함하는 사전 훈련 모델을 생성하는 단계; 및

상기 사전 훈련 모델에 포함되는 상기 손실이 적용된 상기 소스 인코더의 파라미터를 초기 파라미터로 하여 타겟 모델(target model)을 트레이닝하는 단계

를 포함하고,

상기 타겟 뷰에 포함되는 복수의 타겟 패치들은,

상기 소스 뷰에 포함되는 복수의 소스 마스크 토큰들과 겹치지 않도록 마스킹된 복수의 타겟 마스크 토큰들을 포함하고,

동작 방법.

청구항 12

전자 장치에 있어서,

입력 이미지를 기반으로 생성된 소스 뷰(source view) 및 타겟 뷰(target view) 각각을 소스 인코더(source encoder) 및 타겟 인코더(target encoder) 각각에 입력하고, 차별 학습(discrimination learning) 및 상기 입력 이미지에 기반하여 상기 소스 인코더의 출력 및 상기 타겟 인코더의 출력으로부터 손실을 계산하고, 상기 손실이 적용된 상기 소스 인코더를 포함하는 사전 훈련 모델(pre-trained model)을 생성하는 프로세서

를 포함하는, 전자 장치.

청구항 13

제12항에 있어서,

상기 프로세서는,

상기 입력 이미지를 기반으로 상기 소스 뷰를 생성하고, 상기 입력 이미지를 기반으로 상기 타겟 뷰를 생성하고,

상기 소스 뷰는,

상기 입력 이미지가 분할된 복수의 소스 패치들(source patches) 및 상기 복수의 소스 패치들 간의 관계 정보를 포함하는 소스 분류 토큰(source classification token)을 포함하고,

상기 복수의 소스 패치들은,

복수의 소스 이미지 토큰들 및 복수의 소스 마스크 토큰들을 포함하고,

상기 타겟 뷰는,

상기 입력 이미지가 분할된 복수의 타겟 패치들(target patches) 및 상기 복수의 타겟 패치들 간의 관계 정보를 포함하는 타겟 분류 토큰(target classification token)을 포함하고,

상기 복수의 타겟 패치들은,

복수의 타겟 이미지 토큰들 및 복수의 타겟 마스크 토큰들을 포함하는,

전자 장치.

청구항 14

제13항에 있어서,

상기 프로세서는,

토큰화(tokenize)를 통해 상기 입력 이미지를 복수의 패치들로 분할하고, 상기 복수의 패치들 중 일부 패치들에 대해 랜덤 마스킹을 수행하여, 상기 랜덤 마스킹된 상기 복수의 소스 마스크 토큰들 및 상기 랜덤 마스킹되지 않은 상기 복수의 소스 이미지 토큰들을 포함하는 상기 복수의 소스 패치들을 생성하고, 상기 복수의 소스 패치들 간의 관계 정보를 포함하는 상기 소스 분류 토큰을 생성하고, 상기 소스 분류 토큰 및 상기 복수의 소스 패치들을 포함하는 상기 소스 뷰를 생성하는, 전자 장치.

청구항 15

제13항에 있어서,

상기 프로세서는,

토큰화를 통해 상기 입력 이미지를 복수의 패치들로 분할하고, 상기 복수의 타겟 패치들에 대해 상기 소스 뷰의 랜덤 마스킹과 겹치지 않도록 마스킹을 수행하여, 상기 마스킹된 상기 복수의 타겟 마스크 토큰들 및 상기 마스킹되지 않은 상기 복수의 타겟 이미지 토큰들을 포함하는 상기 복수의 타겟 패치들을 생성하고, 상기 복수의 타겟 패치들 간의 관계 정보를 나타내는 상기 타겟 분류 토큰을 생성하고, 상기 타겟 분류 토큰 및 상기 복수의 타겟 패치들을 포함하는 상기 타겟 뷰를 생성하는, 전자 장치.

청구항 16

제13항에 있어서,

상기 프로세서는,

복수의 소스 출력 패치들을 디코더에 입력하여 상기 복수의 소스 마스크 토큰들을 복원하고, 상기 복원된 상기 복수의 소스 마스크 토큰들 각각의 불확실성을 계산하고,

상기 복수의 소스 출력 패치들은,

상기 소스 인코더의 상기 출력 중 상기 복수의 소스 패치들에 대한 출력인,

전자 장치.

청구항 17

제16항에 있어서,

상기 프로세서는,

상기 복원된 상기 복수의 소스 마스크 토큰들 각각과 상기 복수의 소스 마스크 토큰들의 위치에 대응하는 상기 입력 이미지의 패치들 간의 거리인 불확실성을 계산하는, 전자 장치.

청구항 18

제16항에 있어서,

상기 프로세서는,

상기 복수의 소스 출력 패치들 및 복수의 타겟 출력 패치들 중 상기 불확실성이 임계값 미만인 패치들에 대응하는 토큰들을 로컬 차별 학습 모듈에 입력하여 로컬 손실을 계산하고, 소스 출력 분류 토큰 및 타겟 출력 분류 토큰을 글로벌 차별 학습 모듈에 입력하여 글로벌 손실을 계산하고, 상기 복원된 상기 복수의 소스 마스크 토큰들과 상기 입력 이미지를 이용하여 회귀 손실을 계산하고,

상기 복수의 타겟 출력 패치들은,

상기 타겟 인코더의 상기 출력 중 상기 복수의 타겟 패치들에 대응하는 출력이고,

상기 소스 출력 분류 토큰은,

상기 소스 인코더의 출력 중 상기 소스 분류 토큰에 대응하는 출력이고,

상기 타겟 출력 분류 토큰은,

상기 타겟 인코더의 출력 중 상기 타겟 분류 토큰에 대응하는 출력인,

전자 장치.

청구항 19

제18항에 있어서,

상기 프로세서는,

상기 로컬 손실, 상기 글로벌 손실 및 상기 회귀 손실을 상기 소스 인코더에 반영하여 상기 사전 훈련 모델을 생성하는, 전자 장치.

청구항 20

제12항에 있어서,

상기 프로세서는,

상기 사전 훈련 모델에 포함되는 상기 손실이 적용된 상기 소스 인코더의 파라미터를 초기 파라미터로 하여 타겟 모델(target model)을 트레이닝하는, 전자 장치.

발명의 설명

기 술 분 야

[0001] 마스킹된 이미지를 복원하는 사전 훈련 모델을 생성하는 전자 장치 및 그 동작 방법이 개시된다.

배 경 기 술

[0002] 문장에 존재하는 빈칸에 들어갈 단어를 예측하는 자연어 처리 모델인 MLM(masked language modeling)은 높은 확장성(scalability) 및 일반화정도(generalizability)를 가질 수 있다. 이러한 MLM의 특징은 컴퓨터 비전 분야에서 MIM(masked image modeling)의 개발을 촉진했다. MIM은 이미지의 마스킹된 부분을 예측하는 모델일 수 있다. 최근 연구에서 MIM은 인접 픽셀들 간의 로우-레벨 패치 재구성(low-level patch reconstruction)에 집중하여 의미론(semantics) 학습은 부족할 수 있음이 밝혀졌다.

[0003] 전술한 배경기술은 본 개시의 도출과정에서 보유하거나 습득한 것으로서 반드시 본 개시의 출원 전에 일반 공중에 공개된 공지기술이라고 할 수 없다.

발명의 내용

해결하려는 과제

[0004] 본 발명은 서로 겹치지 않도록 반대로 마스킹된 소스 패치들 및 타겟 패치들에 기반한 차별 학습을 수행하여 MIM을 학습하는 방법을 제공한다.

[0005] 본 발명은 학습된 MIM의 파라미터를 이용하여 타겟 모델을 학습하는 방법을 제공한다.

과제의 해결 수단

[0006] 본 발명의 일 실시예에 따른 전자 장치의 동작 방법은, 입력 이미지를 기반으로 생성된 소스 뷰(source view) 및 타겟 뷰(target view) 각각을 소스 인코더(source encoder) 및 타겟 인코더(target encoder) 각각에 입력하는 단계, 차별 학습(discrimination learning) 및 상기 입력 이미지에 기반하여 상기 소스 인코더의 출력 및 상기 타겟 인코더의 출력으로부터 손실을 계산하는 단계, 및 상기 손실이 적용된 상기 소스 인코더를 포함하는 사전 훈련 모델(pre-trained model)을 생성하는 단계를 포함할 수 있다.

[0007] 상기 입력 이미지를 기반으로 상기 소스 뷰를 생성하는 단계, 및 상기 입력 이미지를 기반으로 상기 타겟 뷰를 생성하는 단계를 더 포함하고, 상기 소스 뷰는, 상기 입력 이미지가 분할된 복수의 소스 패치들(source patches) 및 상기 복수의 소스 패치들 간의 관계 정보를 포함하는 소스 분류 토큰(source classification

token)을 포함하고, 상기 복수의 소스 패치들은, 복수의 소스 이미지 토큰들 및 복수의 소스 마스크 토큰들을 포함하고, 상기 타겟 뷰는, 상기 입력 이미지가 분할된 복수의 타겟 패치들(target patches) 및 상기 복수의 타겟 패치들 간의 관계 정보를 포함하는 타겟 분류 토큰(target classification token)을 포함하고, 상기 복수의 타겟 패치들은, 복수의 타겟 이미지 토큰들 및 복수의 타겟 마스크 토큰들을 포함할 수 있다.

[0008] 상기 입력 이미지를 기반으로 상기 소스 뷰를 생성하는 단계는, 토큰화(tokenize)를 통해 상기 입력 이미지를 복수의 패치들로 분할하는 단계, 상기 복수의 패치들 중 일부 패치들에 대해 랜덤 마스킹을 수행하여, 상기 랜덤 마스킹된 상기 복수의 소스 마스크 토큰들 및 상기 랜덤 마스킹되지 않은 상기 복수의 소스 이미지 토큰들을 포함하는 상기 복수의 소스 패치들을 생성하는 단계, 상기 복수의 소스 패치들 간의 관계 정보를 포함하는 상기 소스 분류 토큰을 생성하는 단계, 및 상기 소스 분류 토큰 및 상기 복수의 소스 패치들을 포함하는 상기 소스 뷰를 생성하는 단계를 포함할 수 있다.

[0009] 상기 입력 이미지를 기반으로 상기 타겟 뷰를 생성하는 단계는, 토큰화를 통해 상기 입력 이미지를 복수의 패치들로 분할하는 단계, 상기 복수의 패치들에 대해 상기 소스 뷰의 랜덤 마스킹과 겹치지 않도록 마스킹을 수행하여, 상기 마스킹된 상기 복수의 타겟 마스크 토큰들 및 상기 마스킹되지 않은 상기 복수의 타겟 이미지 토큰들을 포함하는 상기 복수의 타겟 패치들을 생성하는 단계, 상기 복수의 타겟 패치들 간의 관계 정보를 나타내는 상기 타겟 분류 토큰을 생성하는 단계, 및 상기 타겟 분류 토큰 및 상기 복수의 타겟 패치들을 포함하는 상기 타겟 뷰를 생성하는 단계를 포함할 수 있다.

[0010] 복수의 소스 출력 패치들을 디코더에 입력하여 상기 복수의 소스 마스크 토큰들을 복원하는 단계, 및 상기 복원된 상기 복수의 소스 마스크 토큰들 각각의 불확실성을 계산하는 단계를 더 포함하고, 상기 복수의 소스 출력 패치들은, 상기 소스 인코더의 상기 출력 중 상기 복수의 소스 패치들에 대한 출력일 수 있다.

[0011] 상기 복원된 상기 복수의 소스 마스크 토큰들 각각의 상기 불확실성을 계산하는 단계는, 상기 복원된 상기 복수의 소스 마스크 토큰들 각각과 상기 복수의 소스 마스크 토큰들의 위치에 대응하는 상기 입력 이미지의 패치들 간의 거리인 불확실성을 계산하는 단계를 포함할 수 있다.

[0012] 상기 소스 인코더의 출력 및 상기 타겟 인코더의 출력으로부터 손실을 계산하는 단계는, 상기 복수의 소스 출력 패치들 및 복수의 타겟 출력 패치들 중 상기 불확실성이 임계값 미만인 패치들에 대응하는 토큰들을 로컬 차별 학습 모듈에 입력하여 로컬 손실을 계산하는 단계, 소스 출력 분류 토큰 및 타겟 출력 분류 토큰을 글로벌 차별 학습 모듈에 입력하여 글로벌 손실을 계산하는 단계, 및 상기 복원된 상기 복수의 소스 마스크 토큰들과 상기 입력 이미지를 이용하여 회귀 손실을 계산하는 단계를 포함하고, 상기 복수의 타겟 출력 패치들은, 상기 타겟 인코더의 상기 출력 중 상기 복수의 타겟 패치들에 대응하는 출력이고, 상기 소스 출력 분류 토큰은, 상기 소스 인코더의 출력 중 상기 소스 분류 토큰에 대응하는 출력이고, 상기 타겟 출력 분류 토큰은, 상기 타겟 인코더의 출력 중 상기 타겟 분류 토큰에 대응하는 출력일 수 있다.

[0013] 상기 손실이 적용된 상기 소스 인코더를 포함하는 상기 사전 훈련 모델을 생성하는 단계는, 상기 로컬 손실, 상기 글로벌 손실 및 상기 회귀 손실을 상기 소스 인코더에 반영하여 상기 사전 훈련 모델을 생성할 수 있다.

[0014] 상기 사전 훈련 모델에 포함되는 상기 손실이 적용된 상기 소스 인코더의 파라미터를 초기 파라미터로 하여 타겟 모델(target model)을 트레이닝하는 단계를 더 포함할 수 있다.

[0015] 상기 타겟 인코더의 파라미터들이 상기 소스 인코더의 파라미터를 이용한 지수 이동 평균(exponential moving average)을 통해 업데이트되는 단계를 더 포함할 수 있다.

[0016] 본 발명의 일 실시예에 따른 전자 장치의 동작 방법은, 입력 이미지를 기반으로 생성된 소스 뷰 및 타겟 뷰 각각을 소스 인코더 및 타겟 인코더 각각에 입력하는 단계, 차별 학습 및 상기 입력 이미지에 기반하여 상기 소스 인코더의 출력 및 상기 타겟 인코더의 출력으로부터 손실을 계산하는 단계, 상기 손실이 적용된 상기 소스 인코더를 포함하는 사전 훈련 모델을 생성하는 단계, 및 상기 사전 훈련 모델에 포함되는 상기 손실이 적용된 상기 소스 인코더의 파라미터를 초기 파라미터로 하여 타겟 모델(target model)을 트레이닝하는 단계를 포함하고, 상기 타겟 뷰에 포함되는 복수의 타겟 패치들은, 상기 소스 뷰에 포함되는 복수의 소스 마스크 토큰들과 겹치지 않도록 마스킹된 복수의 타겟 마스크 토큰들을 포함할 수 있다.

[0017] 본 발명의 일실시예에 따른 전자 장치는, 입력 이미지를 기반으로 생성된 소스 뷰(source view) 및 타겟 뷰(target view) 각각을 소스 인코더(source encoder) 및 타겟 인코더(target encoder) 각각에 입력하고, 차별 학습(discrimination learning) 및 상기 입력 이미지에 기반하여 상기 소스 인코더의 출력 및 상기 타겟 인코더의 출력으로부터 손실을 계산하고, 상기 손실이 적용된 상기 소스 인코더를 포함하는 사전 훈련 모델(pre-

trained model)을 생성하는 프로세서를 포함할 수 있다.

[0018] 상기 프로세서는, 상기 입력 이미지를 기반으로 상기 소스 뷰를 생성하고, 상기 입력 이미지를 기반으로 상기 타겟 뷰를 생성하고, 상기 소스 뷰는, 상기 입력 이미지가 분할된 복수의 소스 패치들(source patches) 및 상기 복수의 소스 패치들 간의 관계 정보를 포함하는 소스 분류 토큰(source classification token)을 포함하고, 상기 복수의 소스 패치들은, 복수의 소스 이미지 토큰들 및 복수의 소스 마스크 토큰들을 포함하고, 상기 타겟 뷰는, 상기 입력 이미지가 분할된 복수의 타겟 패치들(target patches) 및 상기 복수의 타겟 패치들 간의 관계 정보를 포함하는 타겟 분류 토큰(target classification token)을 포함하고, 상기 복수의 타겟 패치들은, 복수의 타겟 이미지 토큰들 및 복수의 타겟 마스크 토큰들을 포함할 수 있다.

[0019] 상기 프로세서는, 토큰화(tokenize)를 통해 상기 입력 이미지를 복수의 패치들로 분할하고, 상기 복수의 패치들 중 일부 패치들에 대해 랜덤 마스킹을 수행하여, 상기 랜덤 마스킹된 상기 복수의 소스 마스크 토큰들 및 상기 랜덤 마스킹되지 않은 상기 복수의 소스 이미지 토큰들을 포함하는 상기 복수의 소스 패치들을 생성하고, 상기 복수의 소스 패치들 간의 관계 정보를 포함하는 상기 소스 분류 토큰을 생성하고, 상기 소스 분류 토큰 및 상기 복수의 소스 패치들을 포함하는 상기 소스 뷰를 생성할 수 있다.

[0020] 상기 프로세서는, 토큰화를 통해 상기 입력 이미지를 복수의 패치들로 분할하고, 상기 복수의 패치들에 대해 상기 소스 뷰의 랜덤 마스킹과 겹치지 않도록 마스킹을 수행하여, 상기 마스킹된 상기 복수의 타겟 마스크 토큰들 및 상기 마스킹되지 않은 상기 복수의 타겟 이미지 토큰들을 포함하는 상기 복수의 타겟 패치들을 생성하고, 상기 복수의 타겟 패치들 간의 관계 정보를 나타내는 상기 타겟 분류 토큰을 생성하고, 상기 타겟 분류 토큰 및 상기 복수의 타겟 패치들을 포함하는 상기 타겟 뷰를 생성할 수 있다.

[0021] 상기 프로세서는, 복수의 소스 출력 패치들을 디코더에 입력하여 상기 복수의 소스 마스크 토큰들을 복원하고, 상기 복원된 상기 복수의 소스 마스크 토큰들 각각의 불확실성을 계산하고, 상기 복수의 소스 출력 패치들은, 상기 소스 인코더의 상기 출력 중 상기 복수의 소스 패치들에 대한 출력일 수 있다.

[0022] 상기 프로세서는, 상기 복원된 상기 복수의 소스 마스크 토큰들 각각과 상기 복수의 소스 마스크 토큰들의 위치에 대응하는 상기 입력 이미지의 패치들 간의 거리인 불확실성을 계산할 수 있다.

[0023] 상기 프로세서는, 상기 복수의 소스 출력 패치들 및 복수의 타겟 출력 패치들 중 상기 불확실성이 임계값 미만인 패치들에 대응하는 토큰들을 로컬 차별 학습 모듈에 입력하여 로컬 손실을 계산하고, 소스 출력 분류 토큰 및 타겟 출력 분류 토큰을 글로벌 차별 학습 모듈에 입력하여 글로벌 손실을 계산하고, 상기 복원된 상기 복수의 소스 마스크 토큰들과 상기 입력 이미지를 이용하여 회귀 손실을 계산하고, 상기 복수의 타겟 출력 패치들은, 상기 타겟 인코더의 상기 출력 중 상기 복수의 타겟 패치들에 대응하는 출력이고, 상기 소스 출력 분류 토큰은, 상기 소스 인코더의 출력 중 상기 소스 분류 토큰에 대응하는 출력이고, 상기 타겟 출력 분류 토큰은, 상기 타겟 인코더의 출력 중 상기 타겟 분류 토큰에 대응하는 출력일 수 있다.

[0024] 상기 프로세서는, 상기 로컬 손실, 상기 글로벌 손실 및 상기 회귀 손실을 상기 소스 인코더에 반영하여 상기 사전 훈련 모델을 생성할 수 있다.

[0025] 상기 프로세서는, 상기 사전 훈련 모델에 포함되는 상기 손실이 적용된 상기 소스 인코더의 파라미터를 초기 파라미터로 하여 타겟 모델을 트레이닝할 수 있다.

발명의 효과

[0026] 본 발명의 일실시예에 따르면, 서로 겹치지 않게 반대로 마스킹된 소스 패치들 및 타겟 패치들에 기반한 차별 학습을 통해 의미론을 학습한 MIM을 생성할 수 있다.

[0027] 본 발명의 일실시예에 따르면, 사전 학습 모델인 MIM의 파라미터를 이용하여 타겟 모델을 학습할 수 있다.

도면의 간단한 설명

[0028] 도 1은 본 발명의 일 실시예에 따른 전자 장치를 도시한 도면이다.

도 2는 본 발명의 일 실시예에 따른 전자 장치의 동작을 개략적으로 설명하기 위한 도면이다.

도 3은 본 발명의 일 실시예에 따른 사전 학습 모델을 설명하기 위한도면이다.

도 4는 본 발명의 일 실시예에 따른 전자 장치의 동작을 설명하기 위한 플로우 차트이다.

도 5는 본 발명의 일 실시예에 따른 NOM(non-overlapping mask)를 설명하기 위한 도면이다.

도 6 및 도 7은 본 발명의 일 실시예에 따른 소스 뷰 및 타겟 뷰를 생성하는 방법을 설명하기 위한 도면이다.

도 8 및 도 9는 본 발명의 일 실시예에 따른 불확실성을 설명하기 위한 도면이다.

도 10은 본 발명의 일 실시예에 따른 손실 계산을 설명하기 위한 플로우차트이다.

도 11은 본 발명의 일 실시예에 따른 타겟 모델의 성능을 도시한 표이다.

도 12는 본 발명의 일 실시예에 따른 전자 장치의 동작 방법을 설명하기위한 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0029] 이하, 실시예들을 첨부된 도면을 참조하여 상세하게 설명한다. 그러나, 특허출원의 범위가 이러한 실시예들에 의해 제한되거나 한정되는 것은 아니다. 각 도면에 제시된 동일한 참조 부호는 동일한 부재를 나타낸다.

[0030] 아래 설명하는 실시예들에는 다양한 변경이 가해질 수 있다. 아래 설명하는 실시예들은 실시 형태에 대해 한정하려는 것이 아니며, 이들에 대한 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0031] 제1 또는 제2 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 이런 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 이해되어야 한다. 예를 들어, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소는 제1 구성요소로도 명명될 수 있다.

[0032] 실시예에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 실시예를 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수 개의 표현을 포함한다. 본 명세서에서, "A 또는 B", "A 및 B 중 적어도 하나", "A 또는 B 중 적어도 하나", "A, B 또는 C", "A, B 및 C 중 적어도 하나", 및 "A, B, 또는 C 중 적어도 하나"와 같은 문구들 각각은 그 문구들 중 해당하는 문구에 함께 나열된 항목들 중 어느 하나, 또는 그들의 모든 가능한 조합을 포함할 수 있다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 명세서 상에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0033] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 실시예가 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가지고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 가지는 의미와 일치하는 의미를 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0034] 또한, 첨부 도면을 참조하여 설명함에 있어, 도면 부호에 관계없이 동일한 구성 요소는 동일한 참조 부호를 부여하고 이에 대한 중복되는 설명은 생략하기로 한다. 실시예를 설명함에 있어서 관련된 공지 기술에 대한 구체적인 설명이 실시예의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우 그 상세한 설명을 생략한다.

[0035] 이하, 본 발명의 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0037] 도 1은 본 발명의 일 실시예에 따른 전자 장치를 도시한 도면이다.

[0038] 도 1을 참조하면, 전자 장치(100)는 호스트 프로세서(host processor)(110) 및 메모리(120) 및 가속기(accelerator)(130)를 포함할 수 있다. 호스트 프로세서(110), 메모리(120) 및 가속기(130)는 버스(bus), NoC(Network on a Chip), PCIe(Peripheral Component Interconnect Express) 등을 통하여 서로 통신할 수 있다. 도 1에 도시된 전자 장치(100)에는 본 실시예들과 관련된 구성요소들만이 도시되어 있다. 따라서, 전자 장치(100)는 도 1에 도시된 구성요소들 외에 다른 범용적인 구성요소들이 더 포함될 수 있음은 당업자에게 자명하다.

[0039] 호스트 프로세서(110)는 전자 장치(100)를 제어하기 위한 전반적인 기능을 수행하는 역할을 할 수 있다. 호스트 프로세서(110)는 메모리(120)에 저장된 프로그램들 및/또는 명령어들을 실행함으로써, 전자 장치(100)를 전반적으로 제어할 수 있다. 호스트 프로세서(110)는 전자 장치(100) 내에 구비된 CPU(central processing unit), GPU(graphics processing unit), AP(application processor) 등으로 구현될 수 있으나, 이에 제한되지

않는다.

[0040] 메모리(120)는 전자 장치(100) 내에서 처리된 데이터들 및 처리될 데이터들을 저장하는 하드웨어일 수 있다. 또한, 메모리(120)는 전자 장치(100)에 의해 구동될 애플리케이션, 드라이버 등을 저장할 수 있다. 메모리(120)는 DRAM(dynamic random access memory)과 같은 휘발성 메모리(volatile memory) 및/또는 불휘발성 메모리(nonvolatile memory)를 포함할 수 있다.

[0041] 전자 장치(100)는 연산을 위한 하드웨어 가속기(130)를 포함할 수 있다. 가속기(130)는 연산의 특성 상 범용의 호스트 프로세서(110)에서 처리되기 보다는 별도의 전용 프로세서, 다시 말해 가속기(130)에서 처리되는 것이 보다 효율적인 작업들을 처리할 수 있다. 이때 가속기(130)에 포함된 하나 이상의 프로세싱 엘리먼트들(PEs; processing elements)이 활용될 수 있다. 가속기(130)는 예를 들어, 뉴럴 네트워크에 따른 연산을 수행하는 NPU(neural processing unit), TPU(tensor processing unit), DSP(digital signal processor), GPU, Neural Engine 등에 해당될 수 있다.

[0042] 아래에서 설명하는 프로세서는 가속기(130)로 구현될 수 있으나, 실시예가 이에 한정되지 않으며, 프로세서가 호스트 프로세서(110)로 구현될 수도 있다.

[0043] 프로세서는 입력 이미지를 이용하여 소스 뷰(source view) 및 타겟 뷰(target view)를 생성할 수 있다. 소스 뷰는 복수의 소스 패치들(source patches) 및 소스 분류 토큰(source classification token)을 포함할 수 있다. 타겟 뷰는 복수의 타겟 패치들(target patches) 및 타겟 분류 토큰(target classification token)을 포함할 수 있다. 프로세서는 소스 뷰 및 타겟 이미지 각각을 소스 인코더 및 타겟 인코더 각각에 입력할 수 있다.

[0044] 프로세서는 소스 인코더의 출력 및 타겟 인코더의 출력에 기반하여 소스 인코더에 대해 차별 학습을 수행할 수 있다. 프로세서는 소스 인코더의 출력에 기반하여 복수의 소스 마스크 토큰들을 복원하도록 소스 인코더를 학습할 수 있다. 프로세서는 자기 지도 학습을 통해 소스 인코더를 학습할 수 있다.

[0045] 다시 말해, 프로세서는 차별 학습과 마스크 이미지 모델링을 상호 보완적으로 이용할 수 있다. 이를 통해, 프로세서는 의미론을 학습한 MIM인 사전 학습 모델(pre-trained model)을 생성할 수 있다.

[0046] 프로세서는 사전 학습 모델을 이용하여 타겟 모델(예를 들어, 의미론적 분할(semantic segmentation) 및 객체 탐지(object detection) 등)을 학습할 수 있다.

[0047] 이하에서는 전자 장치(100), 구체적으로 프로세서의 동작을 개략적으로 설명하도록 하겠다.

[0049] 도 2는 본 발명의 일 실시예에 따른 전자 장치의 동작을 개략적으로 설명하기 위한 도면이다.

[0050] 프로세서는 토큰화(tokenize)를 통해 입력 이미지(200)으로부터 소스 뷰(210) 및 타겟 뷰(220)를 생성할 수 있다. 입력 이미지(200)는 I 로 나타낼 수 있다. 소스 뷰(210)는 X^s 로 나타낼 수 있다. 타겟 뷰(220)는 X^t 로 나타낼 수 있다. 소스 뷰(210)는 복수의 소스 패치들(211) 및 하나의 소스 분류 토큰(213)을 포함할 수 있다. 복수의 소스 패치들(211) 및 하나의 소스 분류 토큰(213) 각각은 $x^s_{1\cdots N}$ 및 x^s_0 으로 나타낼 수 있다. 다시 말해, 소스 뷰(210)는 $X^s = \{x^s_i\}^N_{i=0}$ 일 수 있다.

[0051] 타겟 뷰(210)는 복수의 타겟 패치들(221) 및 하나의 타겟 분류 토큰(223)을 포함할 수 있다. 복수의 타겟 패치들(221) 및 하나의 타겟 분류 토큰(223) 각각은 $x^t_{1\cdots N}$ 및 x^t_0 로 나타낼 수 있다. 다시 말해 타겟 뷰(210)는 $X^t = \{x^t_i\}^N_{i=0}$ 일 수 있다.

[0052] 복수의 소스 패치들(211)과 복수의 타겟 패치들(221)은 서로 겹치지 않게 반대로 마스킹된 토큰들을 포함할 수 있다. 예를 들어, 복수의 소스 패치들(211)에 포함되는 소스 마스크 토큰들과 복수의 타겟 패치들(221)에 포함되는 타겟 마스크 토큰들은 서로 겹치지지 않을 수 있다.

[0053] 다시 말해, 복수의 소스 패치들(211) 및 복수의 타겟 패치들(221)은 아래의 수학식 1과 같이 정의될 수 있다.

수학식 1

$$x^s = (1-m)x + mx^{\text{MASK}}$$
$$x^t = mx + (1-m)x^{\text{MASK}}$$

[0054]

[0055] M은 다음과 같이 나타낼 수 있는 이진 마스크(binary mask)일 수 있다. $M = \{m_i\}_{i=1}^{N} \in [0,1]$. x^{MASK}는 학습 가능한 마스크 토큰(mask token)일 수 있다. 여기서, N은 토큰의 수(즉, 패치들의 수)일 수 있다. 예를 들어, 입력 이미지가 5x5의 패치로 분할된 경우, N 은 25일 수 있다.

[0056] 즉, 복수의 소스 패치들(211)은 마스킹 되지 않은 복수의 소스 이미지 토큰들 및 마스킹된 복수의 소스 마스크 토큰들을 포함할 수 있다. 복수의 타겟 패치들(221)은 복수의 소스 패치들과 반대로 마스킹된 복수의 타겟 마스크 토큰들 및 마스킹되지 않은 복수의 타겟 이미지 토큰들을 포함할 수 있다. 소스 뷰(210) 및 타겟 뷰(220)를 생성하는 방법에 대해서는 도 5 내지 도 7에서 상세히 설명하도록 하겠다.

[0057] 프로세서는 소스 뷰(210)를 소스 인코더(source encoder)(251)에 입력할 수 있다. 소스 인코더(251)는 $\mathbf{E}^s$로 나타낼 수 있다. 다시 말해, $\mathbf{E}^s$는 자기 지도 설정(self-supervised setup)의 소스 인코더(251)를 나타낼 수 있다. 소스 인코더(251)는 소스 뷰(210)로부터 소스 출력(source output)(230)을 출력할 수 있다. 소스 출력(230)은 Z^s로 나타낼 수 있다. 소스 출력(230)은 복수의 소스 출력 패치들(231) 및 소스 출력 분류 토큰(233)을 포함할 수 있다. 복수의 소스 출력 패치들(231) 및 소스 출력 분류 토큰(233) 각각은 $z^s_{1\ldots N}$ 및 z^s_0로 나타낼 수 있다. 복수의 소스 출력 패치들(231)은 복수의 출력 패치들(211)에 대응하는 출력일 수 있다. 소스 출력 분류 토큰(233)은 소스 분류 토큰(213)에 대응하는 출력일 수 있다.

[0058] 프로세서는 타겟 뷰(220)를 타겟 인코더(target encoder)(253)에 입력할 수 있다.

[0059] 타겟 인코더(253)는 $\mathbf{E}^t$로 나타낼 수 있다. 다시 말해, $\mathbf{E}^t$는 자기 지도 설정의 타겟 인코더(253)를 나타낼 수 있다 타겟 인코더(253)는 타겟 뷰(220)로부터 타겟 출력(source output)(240)을 출력할 수 있다. 타겟 출력(240)은 Z^t로 나타낼 수 있다. 타겟 출력(240)은 복수의 타겟 출력 패치들(241) 및 타겟 출력 분류 토큰(243)을 포함할 수 있다. 복수의 타겟 출력 패치들(241) 및 타겟 출력 분류 토큰(243)은 각각 $z^t_{1\ldots N}$ 및 z^t_0로 나타낼 수 있다. 복수의 타겟 출력 패치들(241)은 복수의 타겟 패치들(221)에 대응하는 출력일 수 있다. 타겟 출력 분류 토큰(243)은 타겟 분류 토큰(223)에 대응하는 출력일 수 있다.

[0060] 즉, 소스 인코더(251)의 출력 및 타겟 인코더(253)의 출력은 아래 수학식 2에 따라 정의될 수 있다.

수학식 2

$$Z^s = \mathbf{E}^s(X^s) \qquad \text{where } Z^s = \{z^s_i\}_{i=0}^{N}$$
$$Z^t = \mathbf{E}^t(X^t) \qquad \text{where } Z^t = \{z^t_i\}_{i=0}^{N}$$

[0061]

[0062] 소스 인코더(251)의 파라미터는 θ^s일 수 있다. 타겟 인코더(253)의 파라미터는 θ^t일 수 있다. 타겟 인코더

(253)의 파라미터 θ^t 는 θ^s 를 이용한 지수 이동 평균을 통해 업데이트될 수 있다.

[0063] 프로세서는 소스 출력(230) 중 복수의 소스 출력 패치들(231)을 디코더(255)에 입력할 수 있다. 디코더(255)는 복수의 소스 마스크 토큰들을 복원할 수 있다. 복원된 복수의 소스 마스크 토큰들은 X^r 으로 나타낼 수 있다. 프로세서는 자기 지도 학습을 수행하므로 입력 이미지가 토큰화된 X 를 이용하여 회귀 손실 및 불확실성을 계산할 수 있다.

[0064] 프로세서는 회귀 손실 계산 모듈(regression loss estimation module)(263)에 복원된 복수의 소스 마스크 토큰들 X^r 및 입력 이미지가 토큰화된 X 를 입력하여 회귀 손실을 계산할 수 있다. 손실을 계산하는 방법에 대해서는 도 10에서 추가로 설명하도록 하겠다.

[0065] 프로세서는 부정확하게 복원된 소스 마스크 토큰들을 제외하여 차별 학습(discrimination learning)을 수행할 수 있다. 따라서, 프로세서는 불확실성 계산 모듈(uncertainty estimation module)(261)을 이용하여 복원된 복수의 소스 마스크 토큰들에 대한 불확실성을 계산할 수 있다. 불확실성을 계산하는 방법에 대해서는 도 8 내지 도 9에서 추가로 설명하도록 하겠다.

[0066] 프로세서는 로컬 차별 학습 모듈(local discrimination learning module)(265)에 복수의 소스 출력 패치들(231) 및 복수의 타겟 출력 패치들(241)을 입력하여 로컬 손실(local loss)을 계산할 수 있다. 프로세서는 글로벌 차별 학습 모듈(global discrimination learning module)(267)에 소스 출력 분류 토큰(233) 및 타겟 출력 분류 토큰(243)을 입력하여 글로벌 손실(global loss)을 계산할 수 있다. 로컬 손실 및 글로벌 손실을 계산하는 방법에 대해서 도 10에서 추가로 설명하겠다.

[0067] 프로세서는 로컬 손실, 글로벌 손실 및 회귀 손실을 이용하여 소스 인코더(251) 및 타겟 인코더(253)를 학습할 수 있다. 다만, 소스 인코더(251)와 타겟 인코더(253)에 동시에 손실을 적용하게 되면 충돌이 발생할 수 있다. 따라서, 프로세스는 지수 이동 평균(exponential moving average, EMA)를 통해 타겟 인코더를 단계적으로 업데이트할 수 있다.

[0068] 프로세서는 학습된 소스 인코더(251)를 포함하는 사전 학습 모델을 생성할 수 있다. 프로세서는 학습된 소스 인코더(251)의 파라미터를 이용하여 타겟 모델(예를 들어, 의미론적 분할 모델 및 객체 탐지 모델 등)을 학습할 수 있다.

[0069] 이하에서는, 사전 학습 모델에 대해서 설명하도록 하겠다.

[0071] 도 3은 본 발명의 일 실시예에 따른 사전 학습 모델을 설명하기 위한도면이다.

[0072] 도 3을 참조하면 사전 학습 모델(300)이 도시된다. 사전 학습 모델(300)은 MIM(masked image modeling) 일 수 있다. 사전 학습 모델(300)은 마스킹된 이미지(310)로부터 복원된 이미지(320)를 생성할 수 있다. 다시 말해, 사전 학습 모델(300)은 마스킹된 이미지(310)를 복원할 수 있다.

[0073] 사전 학습 모델(300)에 기반하여 타겟 모델을 학습하는 경우, 타겟 모델의 정확성이 향상될 수 있다. 사전 학습 모델(300)에 기반하여 타겟 모델을 학습하는 방법은 도 11에서 추가로 설명하겠다.

[0075] 도 4는 본 발명의 일 실시예에 따른 전자 장치의 동작을 설명하기 위한 플로우 차트이다.

[0076] 이하 실시예에서 각 동작들은 순차적으로 수행될 수도 있으나, 반드시 순차적으로 수행되는 것은 아니다. 예를 들어, 각 동작들의 순서가 변경될 수도 있으며, 적어도 두 동작들이 병렬적으로 수행될 수도 있다. 단계(401) 내지 단계(405)는 호스트 프로세서(110)에 의해 수행될 수 있으나 실시예가 이에 한정되지 않으며 가속기(130)으로 구현될 수도 있다.

[0077] 단계(401)에서, 프로세서는 입력 이미지를 기반으로 생성된 소스 뷰 및 타겟 뷰 각각을 소스 인코더 및 타겟 인코더 각각에 입력할 수 있다.

[0078] 프로세서는 입력 이미지를 기반으로 소스 뷰를 생성하는 단계 및 프로세서는 입력 이미지를 기반으로 타겟 뷰를

생성하는 단계를 더 포함할 수 있다.

[0079] 소스 뷰는 입력 이미지가 분할된 복수의 소스 패치들 및 복수의 소스 패치들 간의 관계 정보를 포함하는 소스 분류 토큰을 포함할 수 있다. 복수의 소스 패치들은 복수의 소스 이미지 토큰들 및 복수의 소스 마스크 토큰들을 포함할 수 있다.

[0080] 타겟 뷰는 입력 이미지가 분할된 복수의 타겟 패치들 및 상기 복수의 타겟 패치들 간의 관계 정보를 포함하는 타겟 분류 토큰을 포함할 수 있다. 복수의 타겟 패치들은 복수의 타겟 이미지 토큰들 및 복수의 타겟 마스크 토큰들을 포함할 수 있다.

[0081] 단계(403)에서, 프로세서는 차별 학습 및 입력 이미지에 기반하여 소스 인코더의 출력 및 타겟 인코더의 출력으로부터 손실을 계산할 수 있다.

[0082] 다시 말해, 프로세서는 차별 학습에 기반하여 소스 인코더의 출력 및 타겟 인코더의 출력으로부터 글로벌 손실 및 타겟 손실을 계산할 수 있다. 그리고, 프로세서는 입력 이미지에 기반하여 소스 인코더의 출력으로부터 회귀 손실을 계산할 수 있다.

[0083] 단계(405)에서, 프로세서는 손실이 적용된 소스 인코더를 포함하는 사전 훈련 모델을 생성할 수 있다.

[0084] 이하에서는 소스 뷰 및 타겟 뷰를 생성하는 방법에 대해서 설명하도록 하겠다.

[0086] 도 5는 본 발명의 일 실시예에 따른 NOM(non-overlapping mask)를 설명하기 위한 도면이다.

[0087] 도 5를 참조하면, 차별 학습에서 일반적으로 사용되는 입력 이미지의 증강(augmentation) 중 하나인 랜덤 크롭핑(random cropping)을 설명하기 위한 예시(510)가 도시된다.

[0088] 예시(510)를 참조하면, 입력 이미지(500)를 랜덤으로 크롭핑하여 2개의 부분 뷰가 획득될 수 있다. 랜덤 크롭핑을 통해 공간적으로 다른 부분 뷰가 획득된다는 점에서 랜덤 크롭핑과 마스킹과 동일한 효과를 가질 수 있다. 랜덤 크롭핑 다음으로 마스킹이 수행될 수 있다. 이때, 두개의 부분 뷰들 중 어느 하나에 대해서만 마스킹이 수행될 수 있다. 마스킹은 더 정교하게 부분 뷰를 재생성할 수 있다.

[0089] 다만, 예시(510)을 통해 생성된 부분 뷰들을 이용하여 차별 학습을 수행할 경우 양성 샘플들(positive samples) 간의 차이가 강조되어 글로벌 차별 학습에서 픽셀들 간의 의미론적 관계를 손상시킬 수 있다. 여기서, 양성 샘플들은 랜덤 크롭핑 및 마스킹이 모두 수행된 부분 뷰 및 랜덤 크롭핑만이 수행된 부분 뷰를 포함할 수 있다.

[0090] 예시(520)를 참조하면, 입력 이미지(500)에 대해 겹치지 않도록 마스킹(이하, NOM(non-overlapping mask))이 수행되어 소스 이미지(521) 및 타겟 이미지(523)가 획득될 수 있다. 다시 말해, 소스 이미지(521) 및 타겟 이미지(523)는 서로 겹치지 않게 반대로 마스킹될 수 있다.

[0091] 예를 들어, 소스 이미지(521)에서 (1,3)에 위치한 패치가 마스킹 된 경우, 타겟 이미지(523)에서 (1,3)에 위치한 패치는 마스킹되지 않을 수 있다. 타겟 이미지(523)에서 (1,1)에 위치한 패치가 마스킹 된 경우, 소스 이미지(521)에서 (1,1)에 위치한 패치가 마스킹되지 않을 수 있다.

[0092] 다시 말해, 마스크된 패치를 마스크 토큰이라고 하고, 마스크 되지 않은 패치를 이미지 토큰이라고 할 때, 소스 이미지(521)의 이미지 토큰과 타겟 이미지(523)의 이미지 토큰을 결합하면 입력 이미지(500)를 생성할 수 있다.

[0093] 결국, 입력 이미지(500)에 대해 NOM을 수행함으로써 부분 뷰를 부적절하게 강화하는 것을 피하는 동시에 의미론적 관계를 유지할 수 있는 양성 샘플들을 생성할 수 있다. 여기서, 양성 샘플들은 NOM을 통해 생성된 소스 이미지(521) 및 타겟 이미지(523)을 포함할 수 있다.

[0094] 이하에서는 상술한 NOM을 이용하여 소스 뷰와 타겟 뷰를 생성하는 방법을 설명하도록 하겠다.

[0096] 도 6 및 도 7은 본 발명의 일 실시예에 따른 소스 뷰 및 타겟 뷰를 생성하는 방법을 설명하기 위한 도면이다.

[0097] 이하 실시예에서 각 동작들은 순차적으로 수행될 수도 있으나, 반드시 순차적으로 수행되는 것은 아니다. 예를 들어, 각 동작들의 순서가 변경될 수도 있으며, 적어도 두 동작들이 병렬적으로 수행될 수도 있다. 단계(601) 내지 단계(607) 및 단계(701) 내지 단계(707)는 호스트 프로세서(110)에 의해 수행될 수 있으나 실시예가 이에 한정되지 않으며 가속기(130)으로 구현될 수도 있다.

[0098] 도 6을 참조하면 입력 이미지로부터 소스 뷰를 생성하는 방법이 도시된다.

[0099] 단계(601)에서, 프로세서는 토큰화를 통해 입력 이미지를 복수의 패치들로 분할할 수 있다.

[0100] 예를 들어 프로세서는 토큰화를 통해 입력 이미지를 5x5를 갖는 복수의 패치들로 분할할 수 있다.

[0101] 단계(603)에서, 프로세서는 복수의 패치들 중 일부 패치들에 대해 랜덤 마스킹을 수행하여, 랜덤 마스킹된 복수의 소스 마스크 토큰들 및 랜덤 마스킹되지 않은 복수의 소스 이미지 토큰들을 포함하는 복수의 소스 패치들을 생성할 수 있다.

[0102] 단계(605)에서, 프로세서는 복수의 소스 패치들 간의 관계 정보를 포함하는 소스 분류 토큰을 생성할 수 있다.

[0103] 단계(607)에서, 프로세서는 소스 분류 토큰 및 복수의 소스 패치들을 포함하는 소스 뷰를 생성할 수 있다.

[0104] 도 7을 참조하면, 입력 이미지로부터 타겟 뷰를 생성하는 방법이 도시된다.

[0105] 단계(701)에서, 프로세서는 토큰화를 통해 입력 이미지를 복수의 패치들로 분할할 수 있다.

[0106] 단계(703)에서, 프로세서는 복수의 패치들에 대해 소스 뷰의 랜덤 마스킹과 겹치지 않도록 마스킹을 수행하여, 마스킹된 복수의 타겟 마스크 토큰들 및 마스킹되지 않은 복수의 타겟 이미지 토큰들을 포함하는 복수의 타겟 패치들을 생성할 수 있다.

[0107] 다시 말해서, 프로세서는 도 5에서 설명한 NOM을 수행하여 서로 겹치지 않는 마스크 토큰을 갖는 소스 뷰 및 타겟 뷰를 생성할 수 있다.

[0108] 단계(705)에서, 프로세서는 복수의 타겟 패치들 간의 관계 정보를 나타내는 타겟 분류 토큰을 생성할 수 있다.

[0109] 단계(707)에서, 프로세서는 타겟 분류 토큰 및 복수의 타겟 패치들을 포함하는 타겟 뷰를 생성할 수 있다.

[0110] 즉, NOM만으로도 차별 학습에 적합한 소스 뷰와 타겟 뷰를 생성할 수 있다.

[0111] 이때, 소스 뷰 및 타겟 뷰의 마스킹 비율을 상이할 수 있다. 소스 뷰의 마스킹 비율이 δ^s 일 때, 타겟 뷰의 마스킹 비율 δ^t 는 $1-\delta^s$ 로 나타낼 수 있다. 다시 말해, 소스 뷰 및 타겟 뷰는 비 대칭적인 마스킹 비율을 가질 수 있다.

[0112] 여기서, 소스 뷰의 마스킹 비율은 $\delta^s = \sum_i m/N$ 와 같이 표현될 수 있다. 이때, 타겟 뷰의 마스킹 비율은 $\delta^t = 1-\delta^s = \sum_i \bar{m}/N$ 와 같이 표현될 수 있다. $\bar{m}$와 m의 관계는 $\bar{m} = 1-m$ 와 같이 나타낼 수 있다. 강건한 사전 학습을 위해 $\delta^s > \delta^t$ 가 만족될 수 있다. 다시 말해, 소스 마스크 토큰들의 수가 타겟 마스크 토큰들의 수 보다 많을 수 있다.

[0113] 이하에서는, 부정확하게 복원된 소스 마스크 토큰들을 제거하기 위해 불확실성을 계산하는 방법에 대해 설명하겠다.

[0115] 도 8 및 도 9는 본 발명의 일 실시예에 따른 불확실성을 설명하기 위한 도면이다.

[0116] 의미론적 관계의 불일치로 인한 차별 학습의 불안정성을 피하기 위해 복원된 복수의 소스 마스크 토큰들 중 부정확한 토큰은 제거되어야 할 수 있다. 다시 말해, 복원된 복수의 소스 마스크 토큰들 중 부정확하게 복원된 소스 마스크 토큰들은 차별 학습 시 제외될 수 있다.

[0117] 이하 실시예에서 각 동작들은 순차적으로 수행될 수도 있으나, 반드시 순차적으로 수행되는 것은 아니다. 예를 들어, 각 동작들의 순서가 변경될 수도 있으며, 적어도 두 동작들이 병렬적으로 수행될 수도 있다. 단계(801) 내지 단계(803)는 호스트 프로세서(110)에 의해 수행될 수 있으나 실시예가 이에 한정되지 않으며 가속기(130)으로 구현될 수도 있다.

[0118] 단계(801)에서, 프로세서는 복수의 소스 출력 패치들을 디코더에 입력하여 복수의 소스 마스크 토큰들을 복원할

수 있다.

[0119] 복수의 소스 출력 패치들은 소스 인코더의 출력 중 복수의 소스 패치들에 대한 출력일 수 있다.

[0120] 단계(803)에서, 프로세서는 복원된 복수의 소스 마스크 토큰들 각각의 불확실성을 계산할 수 있다.

[0121] 프로세서는 복원된 복수의 소스 마스크 토큰들 각각과 복수의 소스 마스크 토큰들의 위치에 대응하는 입력 이미지의 패치들 간의 거리인 불확실성을 계산할 수 있다.

[0122] 불확실성은 복원된 복수의 소스 마스크 토큰들 각각과 복수의 소스 마스크 토큰들의 위치에 대응하는 입력 이미지의 패치들 간의 L1 거리(L1 distance)일 수 있다.

[0123] 불확실성은 동일한 위치에 존재하는 복원된 소스 마스크 토큰과 입력 이미지의 패치 간에 계산될 수 있다. 소스 뷰를 생성할 때 토큰화를 통해 입력 이미지가 분할된 패치들의 수와 동일하게 입력 이미지는 분할될 수 있다.

[0124] 예를 들어, 소스 뷰를 생성할 때 토큰화를 통해 입력 이미지가 5x5로 총 25개의 패치로 분할된 경우, 불확실성을 계산하기 위해 입력 이미지도 5x5로 총 25개의 패치로 분할될 수 있다.

[0125] 복원된 소스 마스크 토큰과 해당 복원된 소스 마스크 토큰과 동일한 인덱스를 갖는 입력 이미지의 패치 간의 L1 거리가 계산될 수 있다. 복원된 소스 마스크 토큰과 해당 복원된 소스 마스크 토큰과 동일한 위치를 갖는 입력 이미지의 패치 간의 L1 거리가 계산될 수 있다. 예를 들어, 복원된 소스 마스크 토큰의 인덱스가 (1,3)인 경우, 해당 복원된 소스 마스크 토큰과 (1,3)에 위치한 입력 이미지의 패치 간의 L1 거리가 계산될 수 있다.

[0126] 결국, 간단한 거리 측정을 이용함으로써 별도의 불확실성을 추정하는 네트워크를 사용하지 않을 수 있다. 따라서, 별도의 불확실성을 추정하는 네트워크를 사용하지 않음으로써 계산 비용이 불필요하게 증가되지 않을 수 있다. 다만, 본 개시는 불확실성을 계산하기 위해 더 정교한 별도의 불확실성을 추정하는 네트워크의 사용을 배제하는 것은 아니다.

[0127] 구체적으로, 복원된 복수의 소스 마스크 토큰들 X^r과 손상되지 않은 입력(uncorrupted input) X(즉, 입력 이미지) 간의 발산(divergence)을 측정하여 불확실성 $U = \{u_i\}_{i=1}^{N}$를 모델링하는 불확실성 추정 모듈은 수학식 3과 같이 나타낼 수 있다.

수학식 3

[0128]
$$u = \sigma(m \cdot d(x^r, x)), \qquad \sigma(k) = \frac{1 - e^k}{1 + e^k}$$

[0129] 여기서, $d(\cdot,\cdot)$은 두 입력인 X^r과 X 간의 L1 거리를 나타낼 수 있다. 수학식 3을 통해 불확실성 맵(uncertainty map) u가 획득될 수 있다. 획득된 불확실성 맵 u은 임계값 τ에 의해 $m^u = (u < \tau)$로 절단(truncate)될 수 있다. m^u는 임계값 맵(threshold map)이라고 할 수 있다. m^u는 복원된 복수의 소스 마스크 토큰 중 불확실성이 임계값 τ 미만인 신뢰할 수 있는 토큰을 나타낼 수 있다.

[0130] 임계값 τ는 0.5일 수 있지만, 이에 한정되지 않는다. 임계값 τ가 증가할수록 부정확하게 복원된 소스 마스크 토큰들이 차별 학습에 이용될 수 있다. 이때, 차별 학습이 부정확하게 수행될 수 있다.

[0131] 반면에, 임계값 τ 가 감소할수록 정확하게 복원된 소스 마스크 토큰들이 차별 학습에 이용될 수 있다. 따라서, 차별 학습이 정확하게 수행될 수 있지만, 복수의 소스 마스크 토큰들이 임계값 τ이하로 정확하게 복원될 때까지 차별 학습이 수행되지 않을 수 있어 시간이 지연될 수 있다. 따라서, 임계값 τ는 상황에 따라 적절히 조정되어야 할 수 있다.

[0132] 도 9를 참조하면, 입력 이미지의 패치와 복원된 소스 마스크 토큰 간의 불확실성이 도시된다.

[0133] 토큰들(911, 913, 915, 917)은 복원된 소스 마스크 토큰들일 수 있다. 다시 말해, 디코더를 통해 복원된 소스 마스크 토큰들 일 수 있다. 그라운드 트루스들(ground truths)(901, 903, 905, 907)은 입력 이미지의 패치일 수 있다. 다시 말해, 그라운드 트루스들(901, 903, 905, 907)은 입력 이미지가 분할된 패치들일 수 있다.

[0134] 토큰(911)과 토큰(911)의 위치에 대응하는 입력 이미지의 패치인 그라운드 트루스(901)간의 L1 거리(즉, 불확실성)가 계산될 수 있다. 토큰(913)과 토큰(913)의 위치에 대응하는 입력 이미지의 패치인 그라운드 트루스(903) 간의 L1 거리(즉, 불확실성)가 계산될 수 있다.

[0135] 토큰(911)의 불확실성은 0.2일 수 있다. 토큰(913)의 불확실성은 0.51일 수 있다. 즉, 토큰(913)은 토큰(911)에 비해 부정확하게 복원된 소스 마스크 토큰일 수 있다. 따라서, 임계값이 0.5인 경우, 불확실성이 0.5보다 큰 토큰(913)은 차별 학습에 이용되지 않을 수 있다.

[0136] 마찬가지로, 토큰(915)과 토큰(915)의 위치에 대응하는 입력 이미지의 패치인 그라운드 트루스(905)간의 L1 거리(즉, 불확실성)가 계산될 수 있다. 토큰(917)과 토큰(917)의 위치에 대응하는 입력 이미지의 패치인 그라운드 트루스(907) 간의 L1 거리(즉, 불확실성)가 계산될 수 있다.

[0137] 토큰(915)의 불확실성은 0.29일 수 있다. 토큰(917)의 불확실성은 0.55일 수 있다. 즉, 토큰(917)은 토큰(915)에 비해 부정확하게 복원된 소스 마스크 토큰일 수 있다. 따라서, 임계값이 0.5인 경우, 불확실성이 0.5보다 큰 토큰(913)은 차별 학습에 이용되지 않을 수 있다.

[0138] 이하에서는, 차별 학습 및 회귀 손실을 계산하는 방법에 대해서 설명하겠다.

[0140] 도 10은 본 발명의 일 실시예에 따른 손실 계산을 설명하기 위한 플로우차트이다.

[0141] 프로세서는 소스 인코더의 출력 및 타겟 인코더의 출력으로부터 손실을 계산할 수 있다.

[0142] 이하 실시예에서 각 동작들은 순차적으로 수행될 수도 있으나, 반드시 순차적으로 수행되는 것은 아니다. 예를 들어, 각 동작들의 순서가 변경될 수도 있으며, 적어도 두 동작들이 병렬적으로 수행될 수도 있다. 단계(1001) 내지 단계(1005)는 호스트 프로세서(110)에 의해 수행될 수 있으나 실시예가 이에 한정되지 않으며 가속기(130)로 구현될 수도 있다.

[0143] 단계(1001)에서, 프로세서는 복수의 소스 출력 패치들 및 복수의 타겟 출력 패치들 중 불확실성이 임계값 미만인 패치들에 대응하는 토큰들을 로컬 차별 학습 모듈에 입력하여 로컬 손실을 계산할 수 있다.

[0144] 복수의 소스 출력 패치들은 소스 인코더의 출력 중 복수의 소스 패치들에 대응하는 출력일 수 있다. 복수의 타겟 출력 패치들은 타겟 인코더의 출력 중 복수의 타겟 패치에 대응하는 출력일 수 있다.

[0145] 단계(1002)에서, 프로세서는 소스 출력 분류 토큰 및 타겟 출력 분류 토큰을 글로벌 차별 학습 모듈에 입력하여 글로벌 손실을 계산할 수 있다.

[0146] 소스 출력 분류 토큰은 소스 인코더의 출력 중 소스 분류 토큰에 대응하는 출력일 수 있다. 타겟 출력 분류 토큰은 타겟 인코더의 출력 중 타겟 분류 토큰에 대응하는 출력 일 수 있다.

[0147] 소스 뷰 및 타겟 뷰 간의 차별성을 최대한 활용하기 위해서 이미지 수준(image-level)의 글로벌 차별 학습 및 토큰 수준(token-level)의 로컬 차별 학습이 수행될 수 있다.

[0148] 로컬 차별 학습 모듈과 글로벌 차별 학습 모듈은 소스 및 타겟에 대해 비대칭 헤드(asymmetric head) ϕ^s 및 ϕ^t 를 사용할 수 있다. ϕ^s 는 소스에 대한 비대칭 헤드이고, ϕ^t 은 타겟에 대한 비대칭 헤드일 수 있다.

[0149] ϕ^s 및 ϕ^t 는 모두 토큰 인코딩을 더 작은 잠재 공간(latent space)로 투사(projection) 할 수 있지만, ϕ^s는 추가 예측 헤드(extra prediction head)를 더 가질 수 있다. 여기서, ϕ^t 의 파라미터는 인코더와 마찬가지로 지수 이동 평균(EMA)를 사용하여 ϕ^s 에 의해서 업데이트 될 수 있다.

[0150] 로컬 차별 학습의 비대칭 헤드 $\phi^{s,l}$ 및 $\phi^{t,l}$ 과 글로벌 차별 학습의 비대칭 헤드 $\phi^{s,g}$ 및 $\phi^{t,g}$ 는 정보를 서로 다른 스케일(scale)로 인코딩 하므로 공유되지 않는다.

[0151] 로컬 차별 학습 모듈은 토큰 수준에서 로컬 차별 학습을 수행할 수 있다. 로컬 차별 학습 모듈은 복수의 소스 출력 패치들 및 복수의 타겟 출력 패치들을 입력으로 하되, m^u 로 나타나는 신뢰할 수 있는 토큰만을 이용하여 로컬 손실(local loss)을 계산할 수 있다. 로컬 손실은 수학식 4와 같이 나타낼 수 있다.

수학식 4

[0152]
$$\mathcal{L}_{local} = -\sum_{i=1}^{N} m_i^u \frac{< \phi^{s,l}(z_i^s), \phi^{t,l}(z_i^t) >}{\|\phi^{s,l}(z_i^s)\|_2 \|\phi^{t,l}(z_i^t)\|_2}$$

[0153] 글로벌 차별 학습 모듈은 이미지 수준에서 글로벌 차별 학습을 수행할 수 있다. 글로벌 차별 학습 모듈은 전체 이미지와 관련된 정보를 포함하는 소스 출력 분류 토큰 및 타겟 출력 분류 토큰을 입력으로 하여 글로벌 손실(global loss)을 계산할 수 있다. 로컬 손실은 수학식 5와 같이 나타낼 수 있다.

수학식 5

[0154]
$$\mathcal{L}_{global} = -\delta^u \frac{< \phi^{s,g}(z_0^s), \phi^{t,g}(z_0^t) >}{\|\phi^{s,g}(z_0^s)\|_2 \|\phi^{t,g}(z_0^t)\|_2}$$

[0155] 여기서, 가중치 δ^u 는 $\delta^u = \sum_i m_i^u / \sum_i m_i$ 로 나타낼 수 있다. 이미지 수준에서는 토큰 수준과 달리 불확실성이 임계값 이상인 토큰들에 대한 정보가 포함되어 있을 수 있다. 따라서, 글로벌 차별 학습에서 글로벌 손실은 로컬 손실과 달리 가중치 δ^u 를 배치함으로써 신뢰할 수 있는 토큰들의 비율도 고려해야 할 수 있다.

[0156] 단계(1003)에서, 프로세서는 복원된 복수의 소스 마스크 토큰들과 입력 이미지를 이용하여 회귀 손실을 계산할 수 있다.

[0157] 다시 말해, 프로세서는 디코더에 의해 복원된 복수의 소스 마스크 토큰들과 입력 이미지를 이용하여 L1 회귀 손실(L1 regression loss)을 계산할 수 있다.

[0158] 프로세서는 로컬 손실, 글로벌 손실 및 회귀 손실을 포함하는 전체 손실(overall loss)을 소스 인코더에 반영할 수 있다. 전체 손실은 수학식 6과 같이 나타낼 수 있다.

수학식 6

[0159]
$$\mathcal{L}_{overall} = \mathcal{L}_{local} + \mathcal{L}_{global} + \mathcal{L}_1$$

[0160] 프로세서는 상술한 전체 손실이 적용된 소스 인코더를 포함하는 사전 훈련 모델을 생성할 수 있다. 전체 손실은 소스 인코더뿐만 아니라 타겟 인코더에도 적용될 수 있다. 다만, 전체 손실이 소스 인코더 및 타겟 인코더에 동시에 반영되는 경우 충돌이 발생할 수 있다. 따라서, 타겟 인코더는 지수 이동 평균을 이용하여 업데이트될 수

있다.

[0161] 결론적으로, 상술한 방법을 통해 획득된 사전 훈련 모델의 소스 인코더는 의미론을 학습할 수 있다.

[0162] 이하에서는 전체 손실이 반영된 사전 훈련 모델을 이용하여 타겟 모델을 학습하는 방법 및 타겟 모델의 성능에 대해서 설명하도록 하겠다.

[0164] 도 11은 본 발명의 일 실시예에 따른 타겟 모델의 성능을 도시한 표이다.

[0165] 프로세서는 사전 훈련 모델을 이용하여 타겟 모델을 학습시킬 수 있다. 프로세서는 사전 훈련 모델을 이용한 전이 학습(transfer learning)을 통해 타겟 모델을 학습시킬 수 있다. 프로세서는 사전 훈련 모델에 포함되는 인코더의 파라미터를 이용하여 타겟 모델을 학습시킬 수 있다. 여기서, 사전 훈련 모델에 포함되는 인코더는 전체 손실이 반영된 것일 수 있다.

[0166] 프로세서는 인코더의 파라미터를 타겟 모델의 초기 파라미터로 설정할 수 있다. 프로세서는 설정된 초기 파라미터를 파인 튜닝하여 타겟 모델을 학습시킬 수 있다.

[0167] 이하에서는 설명의 편의를 위해 타겟 모델이 의미론적 분할 모델(semantic segmentation model)인 경우로 가정하겠다. 다만, 상술한 모델 외에 이미지 분류 모델(image classification model) 및 인스턴스 분할 모델(instance segmentation model)과 같은 컴퓨터 비전과 관련된 타겟 모델을 사전 학습된 모델을 이용하여 학습시킬 수 있음은 통상의 기술자에게 자명하다.

[0168] 이하에서는 상술한 방법에 의해 생성된 사전 학습 모델을 기반으로 학습된 타겟 모델을 MUD(MIM with Uncertainty-aware Discrimination)라고 하겠다.

[0169] 도 11을 참조하면 상술한 사전 훈련 모델을 이용하여 학습된 타겟 모델인 의미론적 분할 모델의 성능인 파인 튜닝 mIoU(fine-tuning mIoU)를 나타낸 표(1100)가 도시된다.

[0170] 사전 훈련 에폭(pre-trained epoch)는 사전 훈련의 수행 횟수를 나타낼 수 있다. 예를 들어, 사전 훈련 에폭이 300인 경우, 모든 데이터를 300번 이용하여 학습 했음을 의미할 수 있다.

[0171] 표(1100)을 참조하면, 사전 훈련 에폭이 800 경우 중에서, MUD의 파인 튜닝 mIoU은 50.2로 가장 클 수 있다.

[0172] 또한, 사전 훈련 에폭이 1600인 경우 중에서, MUD의 파인 튜닝 mIoU는 50.5로 가장 클 수 있다.

[0173] 즉, 의미론을 학습한 사전 훈련 모델을 이용하여 타겟 모델인 의미론적 분할 모델을 학습시킬 경우, 성능이 향상될 수 있다.

[0175] 도 12는 본 발명의 일 실시예에 따른 전자 장치의 동작 방법을 설명하기위한 플로우차트이다.

[0176] 이하 실시예에서 각 동작들은 순차적으로 수행될 수도 있으나, 반드시 순차적으로 수행되는 것은 아니다. 예를 들어, 각 동작들의 순서가 변경될 수도 있으며, 적어도 두 동작들이 병렬적으로 수행될 수도 있다. 단계(1201) 내지 단계(1207)는 호스트 프로세서(110)에 의해 수행될 수 있으나 실시예가 이에 한정되지 않으며 가속기(130)로 구현될 수도 있다.

[0177] 단계(1201)에서, 프로세서는 입력 이미지를 기반으로 생성된 소스 뷰 및 타겟 뷰 각각을 소스 인코더 및 타겟 인코더 각각에 입력할 수 있다.

[0178] 타겟 뷰는 복수의 타겟 패치들을 포함할 수 있다. 복수의 타겟 패치들은 소스 뷰에 포함되는 복수의 소스 마스크 토큰들과 겹치지 않도록 마스킹된 복수의 타겟 마스크 토큰들을 포함할 수 있다.

[0179] 단계(1203)에서, 프로세서는 차별 학습 및 입력 이미지에 기반하여 소스 인코더의 출력 및 타겟 인코더의 출력으로부터 손실을 계산할 수 있다.

[0180] 단계(1205)에서, 프로세서는 손실이 적용된 소스 인코더를 포함하는 사전 훈련 모델을 생성할 수 있다.

[0181] 단계(1207)에서, 프로세서는 사전 훈련 모델에 포함되는 손실이 적용된 상기 소스 인코더의 파라미터를 초기 파라미터로 하여 타겟 모델을 트레이닝 할 수 있다.

[0183] 한편, 본 발명에 따른 방법은 컴퓨터에서 실행될 수 있는 프로그램으로 작성되어 마그네틱 저장매체, 광학적 판독매체, 디지털 저장매체 등 다양한 기록 매체로도 구현될 수 있다.

[0184] 본 명세서에 설명된 각종 기술들의 구현들은 디지털 전자 회로조직으로, 또는 컴퓨터 하드웨어, 펌웨어, 소프트웨어로, 또는 그들의 조합들로 구현될 수 있다. 구현들은 데이터 처리 장치, 예를 들어 프로그램가능 프로세서, 컴퓨터, 또는 다수의 컴퓨터들의 동작에 의한 처리를 위해, 또는 이 동작을 제어하기 위해, 컴퓨터 프로그램 제품, 즉 정보 캐리어, 예를 들어 기계 판독가능 저장 장치(컴퓨터 판독가능 매체) 또는 전파 신호에서 유형적으로 구체화된 컴퓨터 프로그램으로서 구현될 수 있다. 상술한 컴퓨터 프로그램(들)과 같은 컴퓨터 프로그램은 컴파일된 또는 인터프리트된 언어들을 포함하는 임의의 형태의 프로그래밍 언어로 기록될 수 있고, 독립형 프로그램으로서 또는 모듈, 구성요소, 서브루틴, 또는 컴퓨팅 환경에서의 사용에 적절한 다른 유닛으로서 포함하는 임의의 형태로 전개될 수 있다. 컴퓨터 프로그램은 하나의 사이트에서 하나의 컴퓨터 또는 다수의 컴퓨터들 상에서 처리되도록 또는 다수의 사이트들에 걸쳐 분배되고 통신 네트워크에 의해 상호 연결되도록 전개될 수 있다.

[0185] 컴퓨터 프로그램의 처리에 적절한 프로세서들은 예로서, 범용 및 특수 목적 마이크로프로세서들 둘 다, 및 임의의 종류의 디지털 컴퓨터의 임의의 하나 이상의 프로세서들을 포함한다. 일반적으로, 프로세서는 판독 전용 메모리 또는 랜덤 액세스 메모리 또는 둘 다로부터 명령어들 및 데이터를 수신할 것이다. 컴퓨터의 요소들은 명령어들을 실행하는 적어도 하나의 프로세서 및 명령어들 및 데이터를 저장하는 하나 이상의 메모리 장치들을 포함할 수 있다. 일반적으로, 컴퓨터는 데이터를 저장하는 하나 이상의 대량 저장 장치들, 예를 들어 자기, 자기-광 디스크들, 또는 광 디스크들을 포함할 수 있거나, 이것들로부터 데이터를 수신하거나 이것들에 데이터를 송신하거나 또는 양쪽으로 되도록 결합될 수도 있다. 컴퓨터 프로그램 명령어들 및 데이터를 구체화하는데 적절한 정보 캐리어들은 예로서 반도체 메모리 장치들, 예를 들어, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(Magnetic Media), CD-ROM(Compact Disk Read Only Memory), DVD(Digital Video Disk)와 같은 광 기록 매체(Optical Media), 플롭티컬 디스크(Floptical Disk)와 같은 자기-광 매체(Magneto-Optical Media), 롬(ROM, Read Only Memory), 램(RAM, Random Access Memory), 플래시 메모리, EPROM(Erasable Programmable ROM), EEPROM(Electrically Erasable Programmable ROM) 등을 포함한다. 프로세서 및 메모리는 특수 목적 논리 회로조직에 의해 보충되거나, 이에 포함될 수 있다.

[0186] 또한, 컴퓨터 판독가능 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 가용매체일 수 있고, 컴퓨터 저장매체 및 전송매체를 모두 포함할 수 있다.

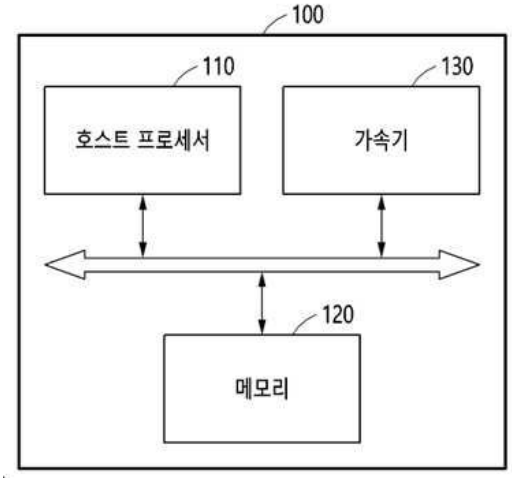
[0187] 본 명세서는 다수의 특정한 구현물의 세부사항들을 포함하지만, 이들은 어떠한 발명이나 청구 가능한 것의 범위에 대해서도 제한적인 것으로서 이해되어서는 안되며, 오히려 특정한 발명의 특정한 실시형태에 특유할 수 있는 특징들에 대한 설명으로서 이해되어야 한다. 개별적인 실시형태의 문맥에서 본 명세서에 기술된 특정한 특징들은 단일 실시형태에서 조합하여 구현될 수도 있다. 반대로, 단일 실시형태의 문맥에서 기술한 다양한 특징들 역시 개별적으로 혹은 어떠한 적절한 하위 조합으로도 복수의 실시형태에서 구현 가능하다. 나아가, 특징들이 특정한 조합으로 동작하고 초기에 그와 같이 청구된 바와 같이 묘사될 수 있지만, 청구된 조합으로부터의 하나 이상의 특징들은 일부 경우에 그 조합으로부터 배제될 수 있으며, 그 청구된 조합은 하위 조합이나 하위 조합의 변형물로 변경될 수 있다.

[0188] 마찬가지로, 특정한 순서로 도면에서 동작들을 묘사하고 있지만, 이는 바람직한 결과를 얻기 위하여 도시된 그 특정한 순서나 순차적인 순서대로 그러한 동작들을 수행하여야 한다거나 모든 도시된 동작들이 수행되어야 하는 것으로 이해되어서는 안 된다. 특정한 경우, 멀티태스킹과 병렬 프로세싱이 유리할 수 있다. 또한, 상술한 실시형태의 다양한 장치 컴포넌트의 분리는 그러한 분리를 모든 실시형태에서 요구하는 것으로 이해되어서는 안되며, 설명한 프로그램 컴포넌트와 장치들은 일반적으로 단일의 소프트웨어 제품으로 함께 통합되거나 다중 소프트웨어 제품에 패키징 될 수 있다는 점을 이해하여야 한다.

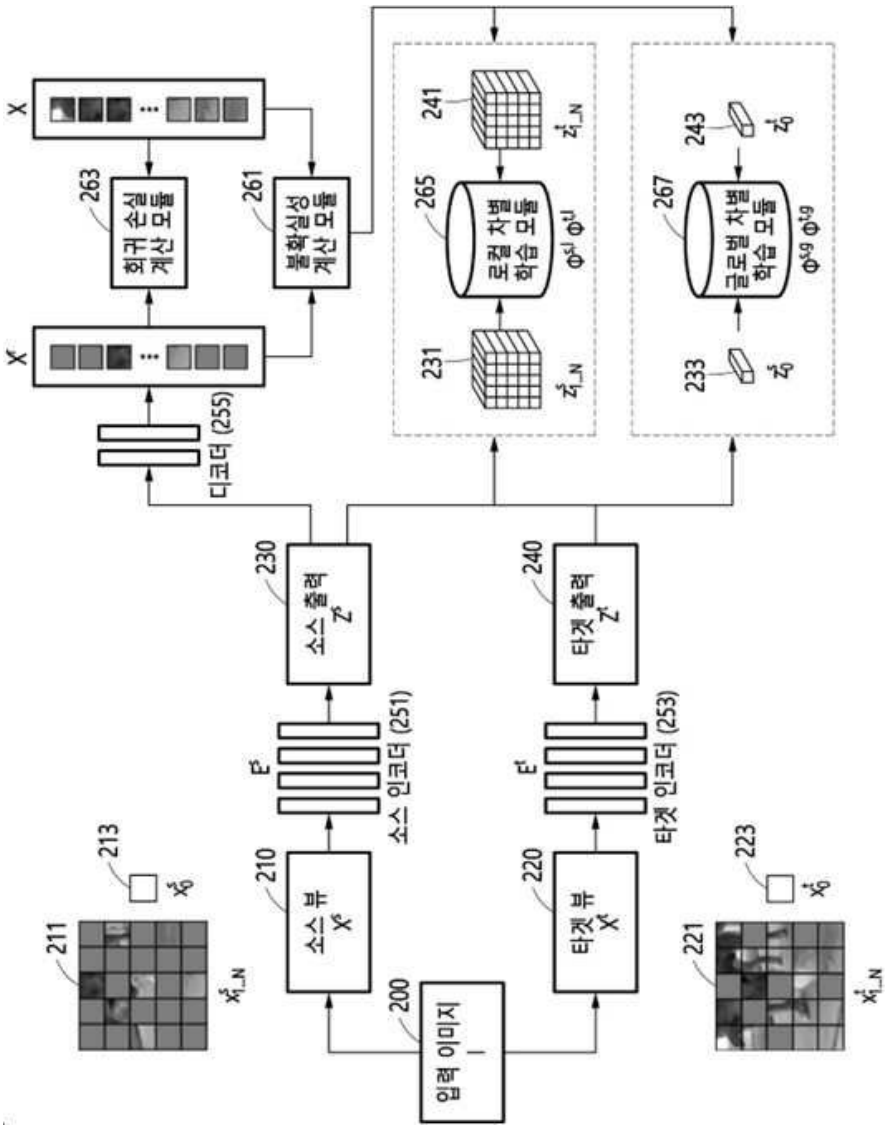
[0189] 한편, 본 명세서와 도면에 개시된 본 발명의 실시 예들은 이해를 돕기 위해 특정 예를 제시한 것에 지나지 않으며, 본 발명의 범위를 한정하고자 하는 것은 아니다. 여기에 개시된 실시 예들 이외에도 본 발명의 기술적 사상에 바탕을 둔 다른 변형 예들이 실시 가능하다는 것은, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 자명한 것이다.

도면

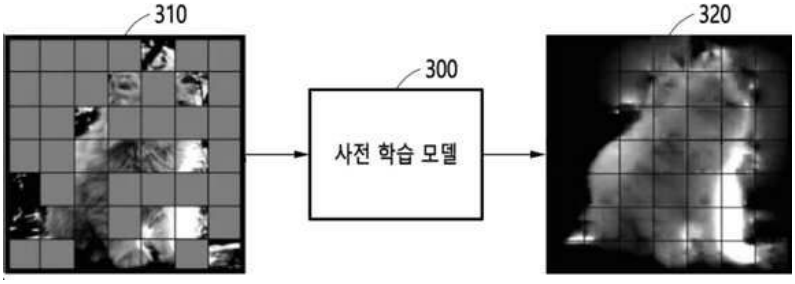
도면1



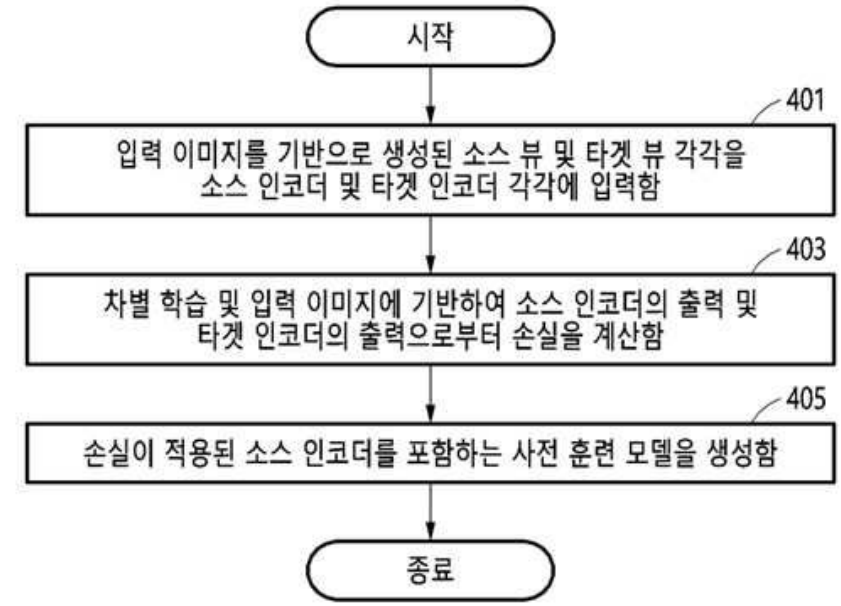
도면2


입력 이미지
200
소스 뷰
210
타겟 뷰
220
소스 인코더 (251)
타겟 인코더 (253)
소스 출력
230
타겟 출력
240
디코더 (255)
회귀 손실 계산 모듈
263
불확실성 계산 모듈
261
로컬 차별 학습 모듈
265
글로벌 차별 학습 모듈
267
241
231
243
233
211
221
213
223

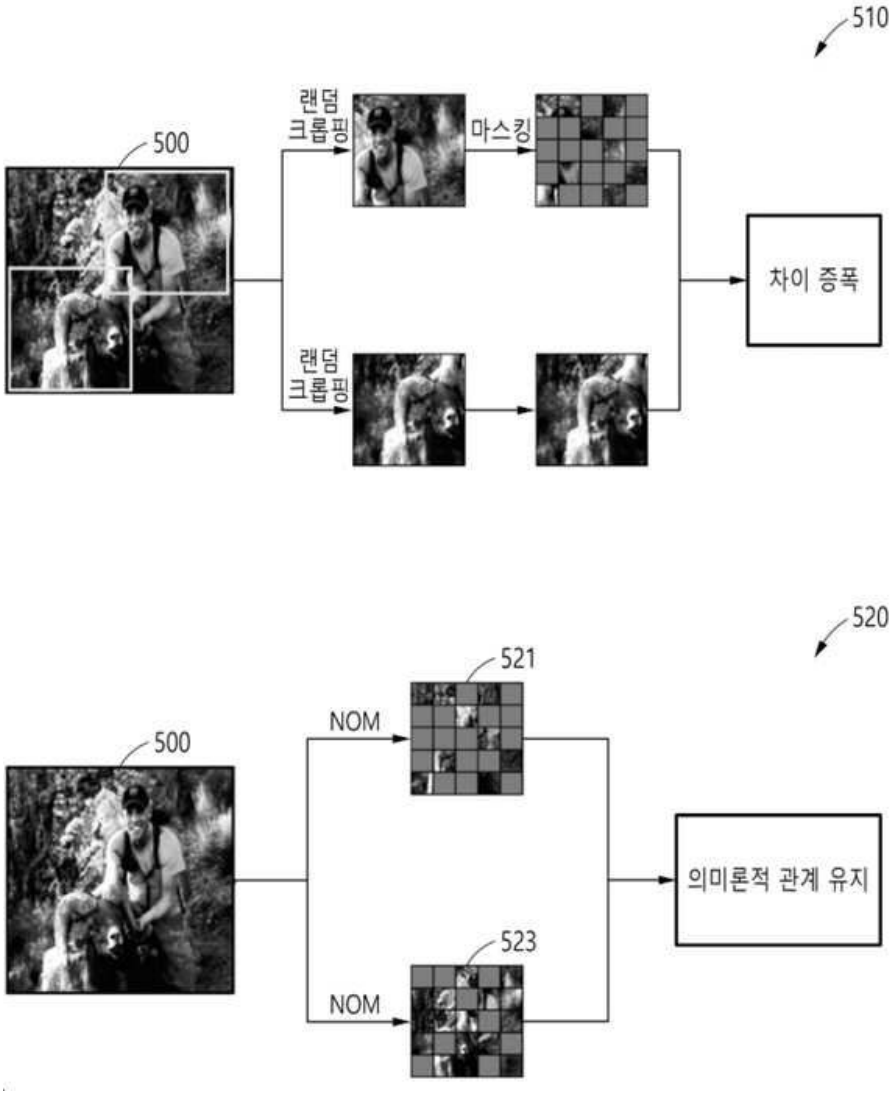
도면3


310
300
사전 학습 모델
320

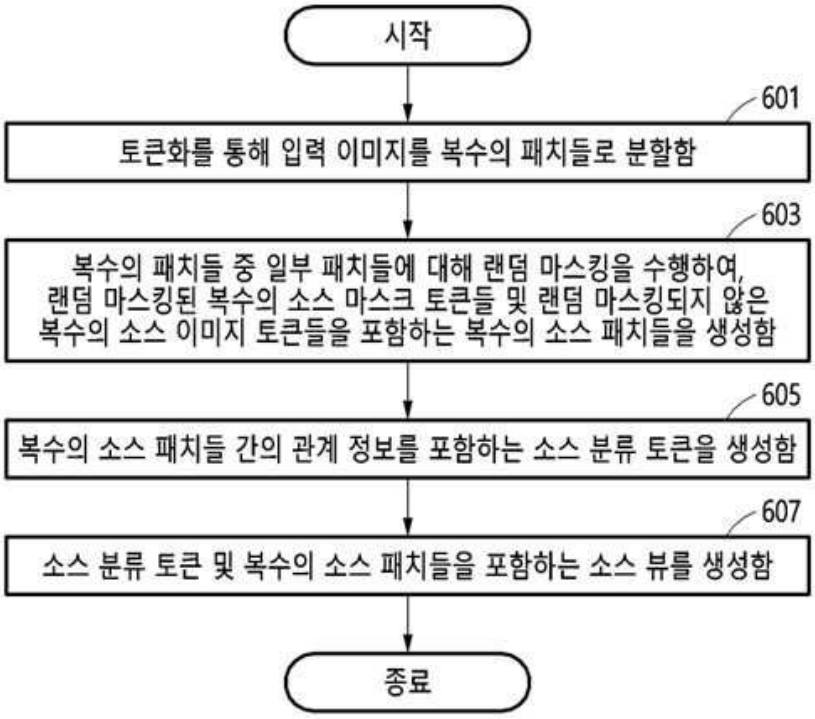
도면4


시작
401
입력 이미지를 기반으로 생성된 소스 뷰 및 타겟 뷰 각각을 소스 인코더 및 타겟 인코더 각각에 입력함
403
차별 학습 및 입력 이미지에 기반하여 소스 인코더의 출력 및 타겟 인코더의 출력으로부터 손실을 계산함
405
손실이 적용된 소스 인코더를 포함하는 사전 훈련 모델을 생성함
종료

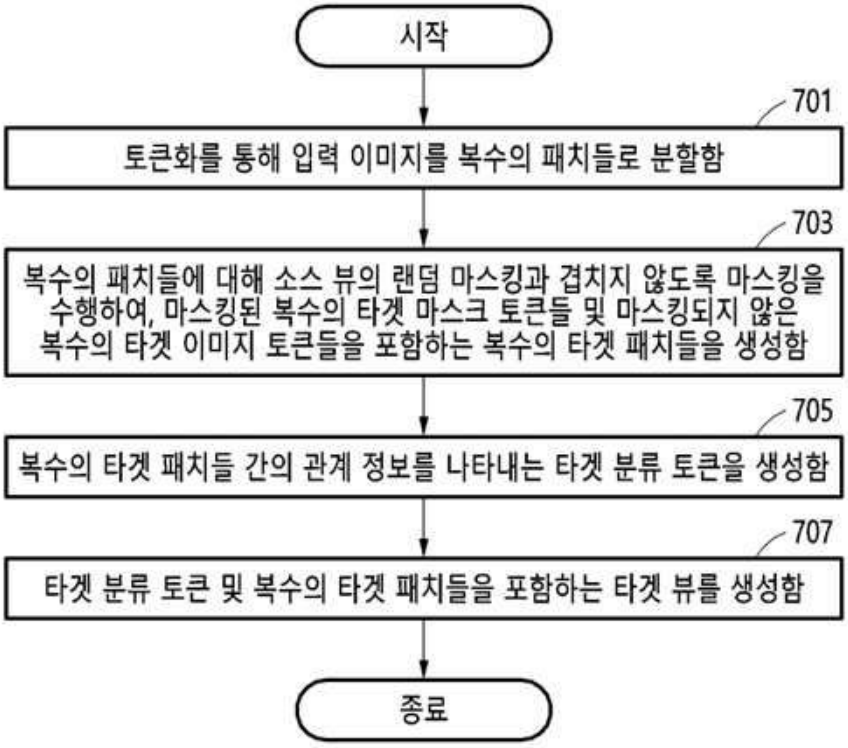
도면5


510
500
랜덤 크롭핑
마스킹
랜덤 크롭핑
차이 증폭
520
500
521
NOM
523
NOM
의미론적 관계 유지

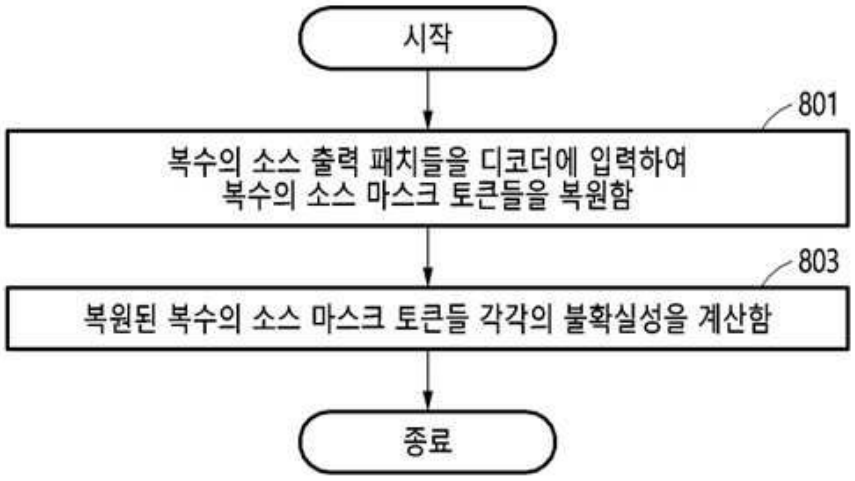
도면6


시작
601
토큰화를 통해 입력 이미지를 복수의 패치들로 분할함
603
복수의 패치들 중 일부 패치들에 대해 랜덤 마스킹을 수행하여, 랜덤 마스킹된 복수의 소스 마스크 토큰들 및 랜덤 마스킹되지 않은 복수의 소스 이미지 토큰들을 포함하는 복수의 소스 패치들을 생성함
605
복수의 소스 패치들 간의 관계 정보를 포함하는 소스 분류 토큰을 생성함
607
소스 분류 토큰 및 복수의 소스 패치들을 포함하는 소스 뷰를 생성함
종료

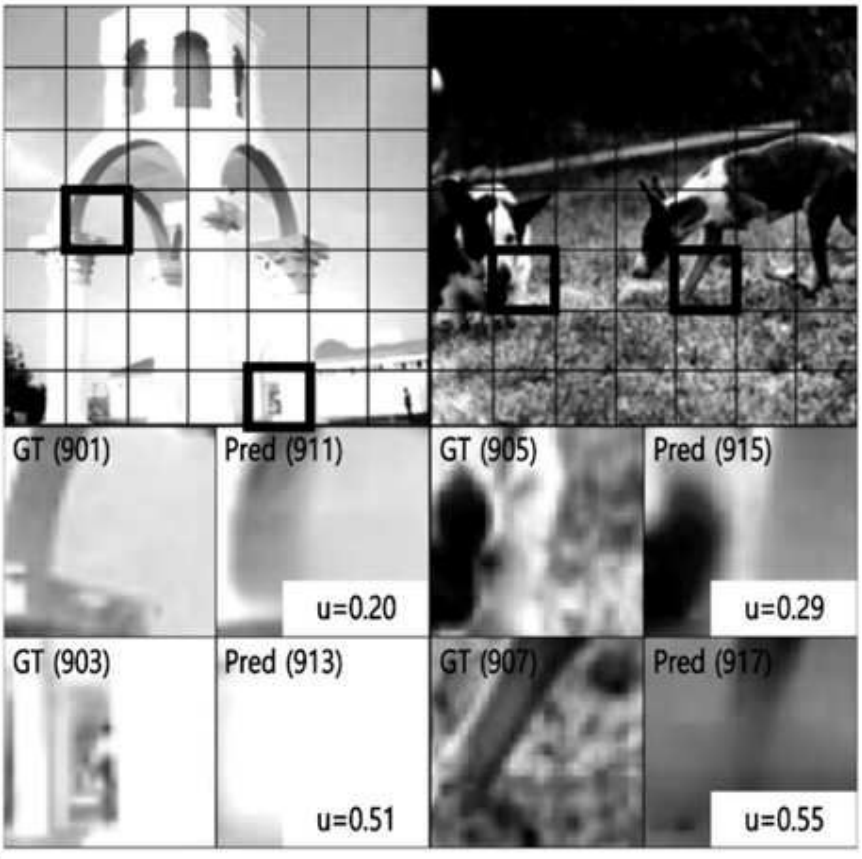
도면7


시작
701
토큰화를 통해 입력 이미지를 복수의 패치들로 분할함
703
복수의 패치들에 대해 소스 뷰의 랜덤 마스킹과 겹치지 않도록 마스킹을 수행하여, 마스킹된 복수의 타겟 마스크 토큰들 및 마스킹되지 않은 복수의 타겟 이미지 토큰들을 포함하는 복수의 타겟 패치들을 생성함
705
복수의 타겟 패치들 간의 관계 정보를 나타내는 타겟 분류 토큰을 생성함
707
타겟 분류 토큰 및 복수의 타겟 패치들을 포함하는 타겟 뷰를 생성함
종료

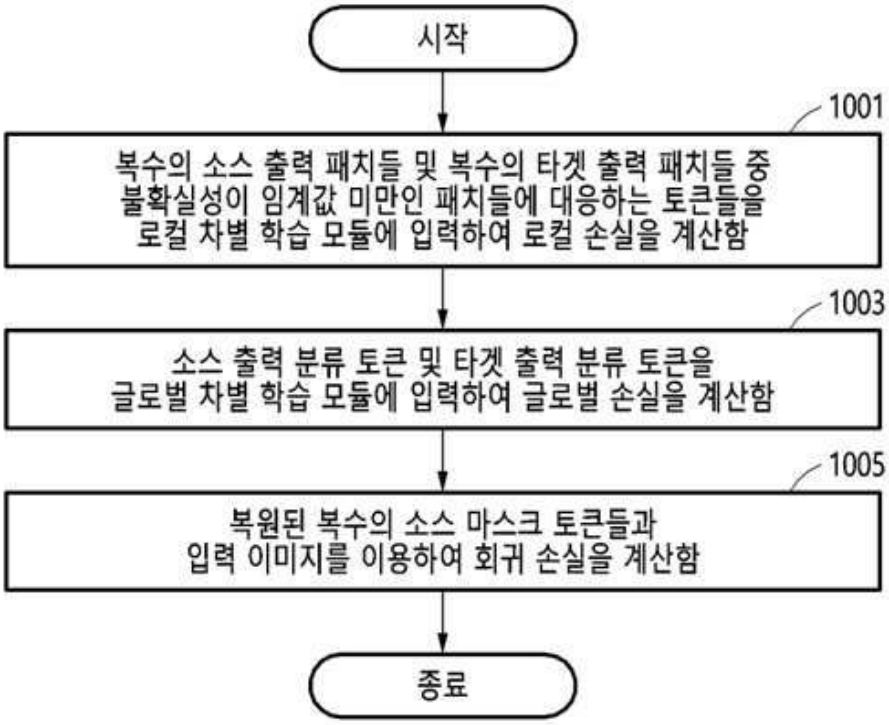
도면8


시작
801
복수의 소스 출력 패치들을 디코더에 입력하여
복수의 소스 마스크 토큰들을 복원함
803
복원된 복수의 소스 마스크 토큰들 각각의 불확실성을 계산함
종료

도면9


GT (901)
Pred (911)
u=0.20
GT (905)
Pred (915)
u=0.29
GT (903)
Pred (913)
u=0.51
GT (907)
Pred (917)
u=0.55

도면10

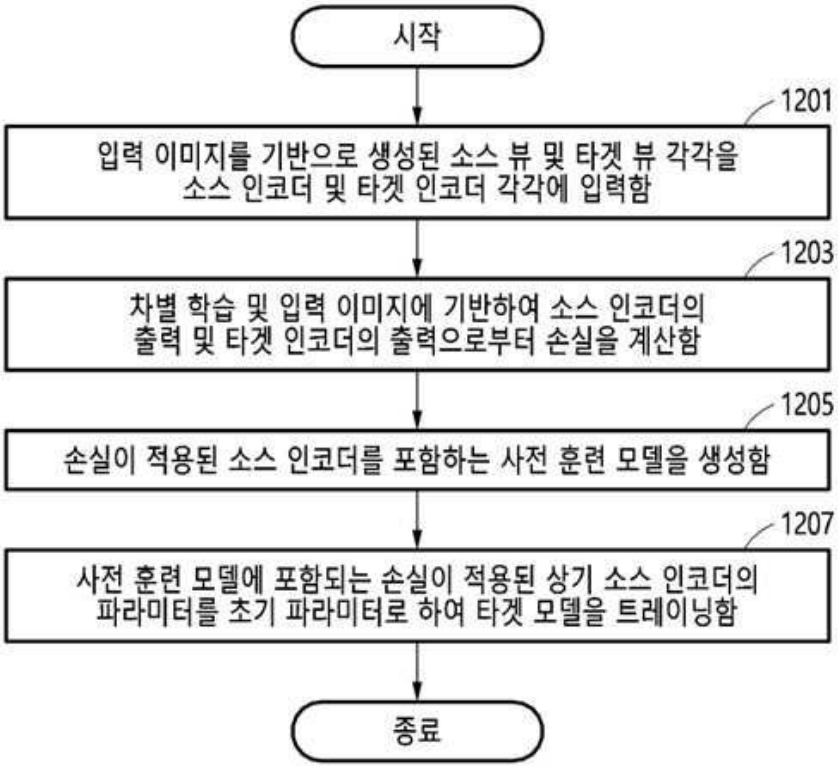


도면11

1100

방법	사전 훈련 에폭	파인 튜닝 mIoU
지도 학습	300	47.4
DINO	1600	46.8
MoCo v3	300	47.3
BEiT	800	47.1
MAE	1600	48.1
CIM	300	43.5
CAE	800	49.7
CAE	1600	50.2
BootMAE	800	49.1
iBOT	1600	50.0
MUD (Ours)	800	50.2
MUD (Ours)	1600	50.5

도면12


시작
1201
입력 이미지를 기반으로 생성된 소스 뷰 및 타겟 뷰 각각을 소스 인코더 및 타겟 인코더 각각에 입력함
1203
차별 학습 및 입력 이미지에 기반하여 소스 인코더의 출력 및 타겟 인코더의 출력으로부터 손실을 계산함
1205
손실이 적용된 소스 인코더를 포함하는 사전 훈련 모델을 생성함
1207
사전 훈련 모델에 포함되는 손실이 적용된 상기 소스 인코더의 파라미터를 초기 파라미터로 하여 타겟 모델을 트레이닝함
종료