



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0150108
(43) 공개일자 2023년10월30일

(51) 국제특허분류(Int. Cl.)
G06T 7/11 (2017.01) G06T 5/50 (2006.01)
G06V 20/40 (2022.01)
(52) CPC특허분류
G06T 7/11 (2017.01)
G06T 5/50 (2023.01)
(21) 출원번호 10-2022-0049656
(22) 출원일자 2022년04월21일
심사청구일자 2022년04월21일

(71) 출원인
한밭대학교 산학협력단
대전광역시 유성구 동서대로 125 (덕명동)
(72) 발명자
장한얼
대전광역시 유성구 동서대로 125,N4동 산업정보관
607호(덕명동)
김원경
대전광역시 서구 동서대로 967, 8동 1001호(내동,
코오롱아파트)
최형욱
대전광역시 유성구 유성대로1689번길 70,KT연구센
타 본관(전민동)
(74) 대리인
특허법인 케이투비

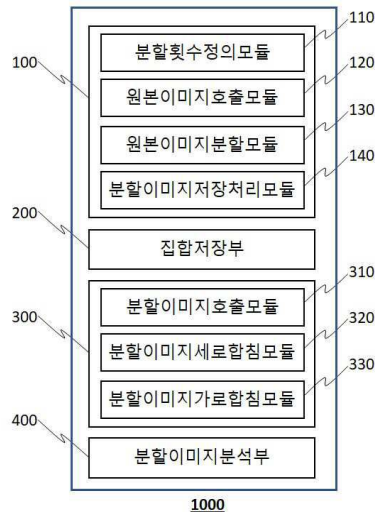
전체 청구항 수 : 총 3 항

(54) 발명의 명칭 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치 및 방법

(57) 요약

본 발명은 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치 및 방법에 관한 것으로서, 더욱 상세하게는 HSR(High Spatial Resolution) 영상의 객체 탐지에서 픽셀 단위의 라벨을 가지고 있는 데이터를 무작위로 선별하여 새로운 데이터를 생성하는 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치 및 방법에 관한 것이다.

대표도 - 도1



(52) CPC특허분류

G06V 20/41 (2022.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1345341784
과제번호	2021RIS-004
부처명	교육부
과제관리(전문)기관명	한국연구재단
연구사업명	지자체-대학협력기반지역혁신사업
연구과제명	(대전세종충남지역혁신플랫폼)충남대학교
기 여 율	1/1
과제수행기관명	(대전세종충남지역혁신플랫폼)충남대학교
연구기간	2021.06.01 ~ 2022.04.30

명세서

청구범위

청구항 1

객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치에 있어서,

원본 데이터셋을 기반으로 4^n 개의 원본 이미지를 선별한 후, 각각 4^n 등분하여 집합저장부(200)에 순서대로 저장해 놓는 분할저장처리부(100)와,

상기 집합저장부(200)에 저장되어 있는 4^{2n} 개의 이미지에서 원본 이미지 당 하나씩 지정한 4^n 개의 이미지를 원본 이미지의 크기와 동일하게 이어붙이는 이어붙임처리부(300)와,

상기 생성된 4^n 분할 이미지를 최종적으로 분석하는 분할이미지분석부(400)를 포함하여 구성되는 것을 특징으로 하는 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치.

청구항 2

제 1항에 있어서,

상기 분할저장처리부(100)는,

사용자에 따라 분할의 횟수를 증가시킬 수 있으며, 분할의 횟수가 증가할수록 데이터의 양을 증가시킴으로써, Semantic Segmentation 성능을 향상시키는 것을 특징으로 하는 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치.

청구항 3

객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 방법에 있어서,

분할저장처리부(100)가 원본 데이터셋을 기반으로 4^n 개의 원본 이미지를 선별한 후, 각각 4^n 등분하여 집합저장부(200)에 순서대로 저장해 놓는 분할저장처리단계(S100)와,

이어붙임처리부(300)가 집합저장부(200)에 저장되어 있는 4^{2n} 개의 이미지에서 원본 이미지 당 하나씩 지정한 4^n 개의 이미지를 원본 이미지의 크기와 동일하게 이어붙이는 이어붙임처리단계(S200);와

분할이미지분석부(400)가 상기 생성된 4^n 분할 이미지를 최종적으로 분석하는 분할이미지분석단계(S300);를 포함하는 것을 특징으로 하는 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 방법.

발명의 설명

기술 분야

[0001]

본 발명은 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치 및 방법에 관한 것으로서, 더욱 상세하게는 HSR(High Spatial Resolution) 영상의 객체 탐지에서 픽셀 단위의 라벨을 가지고 있는 데이터를 무작위로 선별하여 새로운 데이터를 생성하는 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치 및 방법에 관한 것이다.

배경 기술

- [0002] Semantic segmentation 은 이미지 내에 있는 객체들을 픽셀 단위로 분할해 내는 컴퓨터 비전 기술이다.
- [0003] 다시 말해, 이미지의 각 픽셀이 어느 클래스에 속하는지 예측하는 것으로, 서로 다른 종류의 객체들을 분할해 내는 것이 목적이다.
- [0004] 최근 성능을 향상시키기 위해, 심층 신경망(deep neural network)은 hidden layer 를 꾸준히 증가시켜 매개 변수를 늘리는 방식으로 모델의 표현력을 높여왔다.
- [0005] 하지만, 매개 변수를 훈련할 충분한 학습 데이터를 확보하지 않으면 모델의 성능을 저해하는 과적합(overfitting)이 더 쉽게 발생하게 되고, 특히 Semantic Segmentation 은 다른 컴퓨터 비전 작업에 비해 데이터를 확보하는데 있어 많은 비용이 소모된다.
- [0006] 따라서, 충분히 학습하는데 필요한 데이터를 확보하는 기법 중 하나인 증강(Augmentation) 기술이 필요하다.

선행기술문헌

특허문헌

- [0007] (특허문헌 0001) 대한민국등록특허공보 제10-2193257호

발명의 내용

해결하려는 과제

- [0008] 따라서, 본 발명은 상기 종래의 문제점을 해소하기 위해 안출된 것으로,
- [0009] 본 발명의 목적은 데이터의 양이 부족하여 충분히 학습하지 못하는 환경에서, 기존의 원본 데이터 셋을 활용하여 새로운 데이터를 생성함으로써, Semantic Segmentation 성능을 향상시키는데 있다.

과제의 해결 수단

- [0010] 본 발명이 해결하고자 하는 과제를 달성하기 위하여, 본 발명의 일실시예에 따른 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치는,
- [0011] 원본 데이터셋을 기반으로 4^n 개의 원본 이미지를 선별한 후, 각각 4^n 등분하여 집합저장부(200)에 순서대로 저장해 놓는 분할저장처리부(100)와,
- [0012] 상기 집합저장부(200)에 저장되어 있는 4^{2n} 개의 이미지에서 원본 이미지 당 하나씩 지정한 4^n 개의 이미지를 원본 이미지의 크기와 동일하게 이어붙이는 이어붙임처리부(300)와,
- [0013] 상기 생성된 4^n 분할 이미지를 최종적으로 분석하는 분할이미지분석부(400)를 포함하여 구성됨으로써, 본 발명의 과제를 해결하게 된다.

발명의 효과

- [0014] 본 발명에 따른 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치는,
- [0015] 기존의 데이터를 활용하여 새로운 데이터를 생성할 수 있으며, 분할 횟수에 따라 기존 데이터의 수보다 더 많은 데이터를 확보할 수 있다.

도면의 간단한 설명

- [0016] 도 1은 본 발명의 일실시예에 따른 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치의 구성도이다.
- 도 2는 본 발명의 일실시예에 따른 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치의 데이터 증강 예시를 간략하게 설명한 개념도이다.
- 도 3은 본 발명의 일실시예에 따른 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 방법을 나타낸 흐름도이다.

도 4는 본 발명의 일실시예에 따른 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 방법의 4분할 이미지 생성 과정을 나타낸 흐름도이며, 도 5는 생성된 4분할 이미지를 나타낸 예시도이며, 도 6은 Data Augmentation 이전 시각화 이미지와 베이스 라인의 Data Augmentation 과정에서 Random Crop으로 인해 4 분할 이미지는 원본 이미지와의 차별점이 없음을 나타낸 예시도이다.

도 7은 본 발명의 일실시예에 따른 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 방법의 16분할 이미지 생성 과정을 나타낸 흐름도이며, 도 8은 생성된 16분할 이미지와 16분할 이미지를 추가한 실험 결과값을 나타낸 예시도이다.

도 9는 본 발명의 일실시예에 따른 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 방법의 64분할 이미지 생성 과정을 나타낸 흐름도이며, 도 10은 생성된 64분할 이미지와 64분할 이미지를 추가한 실험 결과값을 나타낸 예시도이며, 도 11은 동일 설정, baseline과 64분할 이미지를 추가한 결과와의 비교 및 최종 시각화 결과를 나타낸 예시도이다.

발명을 실시하기 위한 구체적인 내용

- [0017] 본 발명의 과제를 해결하기 위하여 바람직한 실시예에 따른 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치는,
- [0018] 원본 데이터셋을 기반으로 4^n 개의 원본 이미지를 선별한 후, 각각 4^n 등분하여 집합저장부(200)에 순서대로 저장해 놓는 분할저장처리부(100)와,
- [0019] 상기 집합저장부(200)에 저장되어 있는 4^{2n} 개의 이미지에서 원본 이미지 당 하나씩 지정한 4^n 개의 이미지를 원본 이미지의 크기와 동일하게 이어붙이는 이어붙임처리부(300)와,
- [0020] 상기 생성된 4^n 분할 이미지를 최종적으로 분석하는 분할이미지분석부(400)를 포함하여 구성되는 것을 특징으로 한다.
- [0021] 이때, 상기 분할저장처리부(100)는,
- [0022] 원본 이미지의 분할 횟수를 정하는 분할횟수정의모듈(110);과
- [0023] 분할할 원본 이미지를 불러오는 원본이미지호출모듈(120);과
- [0024] 상기 호출된 원본 이미지를 정의한 횟수만큼 분할하는 원본이미지분할모듈(130);과
- [0025] 상기 분할된 이미지를 집합저장부(200)에 저장 처리하는 분할이미지저장처리모듈(140);을 포함하는 것을 특징으로 한다.
- [0026] 이때, 상기 이어붙임처리부(300)는,
- [0027] 일정한 순서로 저장된 분할된 이미지를 불러오는 분할이미지호출모듈(310);과
- [0028] 상기 불러온 분할된 이미지를 세로로 합치는 분할이미지세로합침모듈(320);과
- [0029] 상기 합쳐진 이미지를 가로로 합치는 분할이미지가로합침모듈(330);을 포함하는 것을 특징으로 한다.
- [0030] 이때, 상기 분할이미지분석부(400)는,
- [0031] 기존 원본 이미지와 4^n 분할 이미지를 포함한 데이터 셋을 입력 데이터로 사용해 분류 모델을 통해 판별하는 분류모델판별모듈(410);을 포함하는 것을 특징으로 한다.
- [0032] 이때, 상기 분할저장처리부(100)는,
- [0033] 사용자에게 따라 분할의 횟수를 증가시킬 수 있으며, 분할의 횟수가 증가할수록 데이터의 양을 증가시킴으로써, Semantic Segmentation 성능을 향상시키는 것을 특징으로 한다.
- [0034] 한편, 본 발명인 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 방법은,
- [0035] 분할저장처리부(100)가 원본 데이터셋을 기반으로 4^n 개의 원본 이미지를 선별한 후, 각각 4^n 등분하여 집합저장부(200)에 순서대로 저장해 놓는 분할저장처리단계(S100)와,

- [0036] 이어붙임처리부(300)가 집합저장부(200)에 저장되어 있는 4^{2n} 개의 이미지에서 원본 이미지 당 하나씩 지정한 4^n 개의 이미지를 원본 이미지의 크기와 동일하게 이어붙이는 이어붙임처리단계(S200);와
- [0037] 분할이미지분석부(400)가 상기 생성된 4^n 분할 이미지를 최종적으로 분석하는 분할이미지분석단계(S300);를 포함하는 것을 특징으로 한다.
- [0038] 이하, 본 발명에 의한 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치 및 방법의 실시예를 통해 상세히 설명하도록 한다.
- [0039] 도 1은 본 발명의 일실시예에 따른 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치의 구성도이다.
- [0040] 도 1에 도시한 바와 같이, 본 발명인 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치(1000)는 크게, 분할저장처리부(100), 집합저장부(200), 이어붙임처리부(300), 분할이미지분석부(400)를 포함하여 구성되게 된다.
- [0041] 구체적으로 설명하자면, 상기 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 장치(1000)는 영상에서 객체를 탐지하는 데 있어, 픽셀 단위의 라벨을 가지고 있는 데이터를 무작위로 선별하여 새로운 데이터를 생성하는 기술을 제공하게 된다.
- [0042] 즉, 라벨을 가진 원본 데이터 셋에서 4^n 개의 이미지를 무작위로 선별해 각 이미지를 4^n 등분하여 순서대로 집합저장부에 저장한 후, 집합에서 규칙에 따라 순서번호($n*1 \sim n$)에 해당하는 이미지를 불러와 새로운 이미지를 생성함으로써, 데이터의 양을 증대시킨다.
- [0043] 이를 통해 원본 이미지를 활용하여 4^n 개의 추가 이미지를 생성할 수 있으며, Semantic Segmentation 성능 향상으로 이어진다.
- [0044] 상기과 같은 기능을 제공하기 위하여, 분할저장처리부(100)는 원본 데이터셋을 기반으로 4^n 개의 원본 이미지를 선별한 후, 각각 4^n 등분하여 집합저장부(200)에 순서대로 저장해 놓는 기능을 수행하게 된다.
- [0045] 도 2의 경우, 4 개의 원본 이미지를 선별하였고, 각각 4 등분하여 집합저장부에 저장하게 되는 것이다.
- [0046] 상기과 같은 기능을 수행하기 위하여, 상기 분할저장처리부(100)는,
- [0047] 원본 이미지의 분할 횟수를 정하는 분할횟수정의모듈(110);과
- [0048] 분할할 원본 이미지를 불러오는 원본이미지호출모듈(120);과
- [0049] 상기 호출된 원본 이미지를 정의한 횟수만큼 분할하는 원본이미지분할모듈(130);과
- [0050] 상기 분할된 이미지를 집합저장부(200)에 저장 처리하는 분할이미지저장처리모듈(140);을 포함하는 것을 특징으로 한다.
- [0051] 구체적으로 상기 분할횟수정의모듈(110)은 원본 이미지의 분할 횟수를 정하는 것이다.
- [0052] 예를 들어, 4 회를 설정하게 되는 것이다.
- [0053] 상기 원본이미지호출모듈(120)은 분할할 원본 이미지를 불러오게 되는데, 도 2의 경우, 이미지 A를 호출하게 되는 것이다.
- [0054] 상기 원본이미지분할모듈(130)은 상기 호출된 원본 이미지를 정의한 횟수만큼 분할하는 기능을 수행하게 된다.
- [0055] 도 2의 경우, 이미지 A를 정의한 횟수인 4로 분할하게 되는 것이다.
- [0056] 상기 분할이미지저장처리모듈(140)은 상기 분할된 이미지를 집합저장부(200)에 저장 처리하는 것이다.
- [0057] 그리고, 상기 이어붙임처리부(300)는 상기 집합저장부(200)에 저장되어 있는 4^{2n} 개의 이미지에서 원본 이미지 당 하나씩 지정한 4^n 개의 이미지를 원본 이미지의 크기와 동일하게 이어붙이는 기능을 수행하게 된다.
- [0058] 상기과 같은 기능을 수행하기 위하여, 상기 이어붙임처리부(300)는,

- [0059] 일정한 순서로 저장된 분할된 이미지를 불러오는 분할이미지호출모듈(310);과
- [0060] 상기 불러온 분할된 이미지를 세로로 합치는 분할이미지세로합침모듈(320);과
- [0061] 상기 합쳐진 이미지를 가로로 합치는 분할이미지가로합침모듈(330);을 포함하는 것을 특징으로 한다.
- [0062] 구체적으로, 상기 분할이미지호출모듈(310)은 일정한 순서로 저장된 분할된 이미지를 불러오는 기능을 수행하게 된다.
- [0063] 예를 들어, 도 2의 경우, 이미지 A일 경우에 이미지 A1, 이미지 A2, 이미지 A3, 이미지 A4를 불러오게 되는 것이다.
- [0064] 이때, 상기 분할이미지세로합침모듈(320)은 상기 불러온 분할된 이미지를 세로로 합치는 것이며, 분할이미지가로합침모듈(330)은 상기 합쳐진 이미지를 가로로 합치는 것이다.
- [0065] 예를 들어, 도 6의 경우와 같이 세로로 합치고, 가로로 합쳐진 이미지를 제공하게 되는 것이다.
- [0066] 그리고, 상기 분할이미지분석부(400)는 상기 생성된 4^n 분할 이미지를 최종적으로 분석하는 기능을 수행하게 된다.
- [0067] 상기 기능을 수행하기 위하여, 상기 분할이미지분석부(400)는,
- [0068] 기존 원본 이미지와 4^n 분할 이미지를 포함한 데이터 셋을 입력 데이터로 사용해 분류 모델을 통해 판별하는 분류모델판별모듈(410);을 포함하는 것을 특징으로 한다.
- [0069] 예를 들어, 데이터를 증대시키는 방법이기여, Semantic Segmentation를 수행하는 데 있어 기존의 Fully convolutional structure를 가진 모든 네트워크 모델에서 사용할 수 있다.
- [0070] 사용 가능한 대표적인 모델로는 FCN, DeepLab V3+, UNet, FarSeg, PANet, PointFlow 등의 딥러닝 분류 모델이 있다.
- [0071] 한편, 본 발명인 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 방법은,
- [0072] 분할저장처리부(100)가 원본 데이터셋을 기반으로 4^n 개의 원본 이미지를 선별한 후, 각각 4^n 등분하여 집합저장부(200)에 순서대로 저장해 놓는 분할저장처리단계(S100)와,
- [0073] 이어붙임처리부(300)가 집합저장부(200)에 저장되어 있는 4^{2n} 개의 이미지에서 원본 이미지 당 하나씩 지정한 4^n 개의 이미지를 원본 이미지의 크기와 동일하게 이어붙이는 이어붙임처리단계(S200);와
- [0074] 분할이미지분석부(400)가 상기 생성된 4^n 분할 이미지를 최종적으로 분석하는 분할이미지분석단계(S300);를 포함하는 것을 특징으로 한다.
- [0075] 구체적으로, 도 3에 도시한 바와 같이, 객체 탐지를 위한 분할 및 통합 방식의 데이터 증강 방법은 분할저장처리부(100)가 원본 데이터셋을 기반으로 4^n 개의 원본 이미지를 선별한 후, 각각 4^n 등분하여 집합저장부(200)에 순서대로 저장해 놓게 된다.(S100)
- [0076] 이후, 이어붙임처리부(300)가 집합저장부(200)에 저장되어 있는 4^{2n} 개의 이미지에서 원본 이미지 당 하나씩 지정한 4^n 개의 이미지를 원본 이미지의 크기와 동일하게 이어붙이게 되는 것이다.(S200)
- [0077] 이후, 분할이미지분석부(400)가 상기 생성된 4^n 분할 이미지를 최종적으로 분석하는 것이다.(S300)
- [0078] 상기과 같은 방법을 통해, 기존의 데이터를 활용하여 새로운 데이터를 생성할 수 있으며, 분할 횟수에 따라 기존 데이터의 수보다 더 많은 데이터를 확보할 수 있는 효과를 제공하게 된다.
- [0079] 구체적으로 설명하자면, Semantic Segmentation 훈련 데이터는 픽셀 수준의 라벨링 절차가 필요하며, 이는 다른 비전 작업에 비해 훨씬 많은 시간과 비용이 소모되는 문제점을 가진다.
- [0080] 충분한 데이터를 확보하지 못하면 저조한 성능을 보이거나, 적은 데이터의 양으로 인해 데이터에 포함된 노이즈를 학습하는 등 불필요할 정도로 학습하게 된다.

- [0081] 즉, 훈련 데이터 외의 새로운 데이터에 대한 재성능을 발휘하지 못하는 과적합(overfitting)이 발생하게 된다.
- [0082] 본 발명을 통해 기존의 데이터를 활용하여 새로운 데이터를 생성할 수 있으며, 분할 횟수에 따라 기존 데이터의 수보다 더 많은 데이터를 확보할 수 있게 되는 것이다.
- [0083] 최근 CPS와 같이 라벨을 가지지 않은 이미지를 추가 데이터로 사용하여 학습시키는 Semi-supervised Semantic Segmentation이 연구되고 있지만, 이는 충분한 성능을 확보하지 못했을 때 저조한 개선 혹은 성능 하락을 야기한다.
- [0084] 그러기 때문에, 새롭게 생성된 4^n 분할 이미지는 데이터가 부족하다는 본질적인 문제를 해결할 수 있다.
- [0085] 충분한 데이터를 확보함으로써 모델이 일반적인 패턴을 학습하게 되어 과적합(overfitting) 발생을 줄일 수 있으며, 실험 결과, baseline 대비 mIoU 8%정도의 성능 향상을 제공할 수 있었다.
- [0086] 하지만 분할 횟수가 증가할수록 데이터가 복잡한 형태를 가지게 되고, 초당 분할 이미지 수인 FPS가 감소하게 된다.
- [0087] 그래서, 중요시 하는 성능 지표에 따라 분할 횟수를 설정해야 한다.
- [0088] 또한, 영상에서 건물과 도로를 검출하는 실험 환경과 달리 Sementic Segmentation은 더 다양한 객체에 대해 예측을 수행한다.
- [0089] 건물과 도로와 달리 분할을 수행했을 때 객체의 특징을 훼손할 수 있고, 이는 부정적인 영향을 야기할 수 있다.

[0090] **실험 환경 및 결과**

[0091] 1. 데이터 셋

표 1

Class	Train	Validation
3개(배경 포함)	212개	212개

[0093] 영상 이미지로서, 2개의 객체 클래스(road, build)와 1개의 배경 클래스를 가진다.

[0094] 2. 공통된 설정

표 2

Model Architecture	UNet++
Optimizer	Adam
Loss	DiceLoss
Evaluation Metric	mIoU

[0096] learning rate는 0.001, epoch는 10으로 설정된다.

[0097] 입력 이미지에 대해 무작위 Crop, 뒤집기 등 간단한 Data Augmentation 을 수행한다.

[0098] 3. 차별된 설정(4^n 분할 이미지)

표 3

16분할 이미지	64분할 이미지
1012개	1012개

[0100] 4. 결과

표 4

분류	갯수	mIoU
Baseline	212	69.5%
16분할	212+1012	75.6%
64분할	212+1012	77.7%

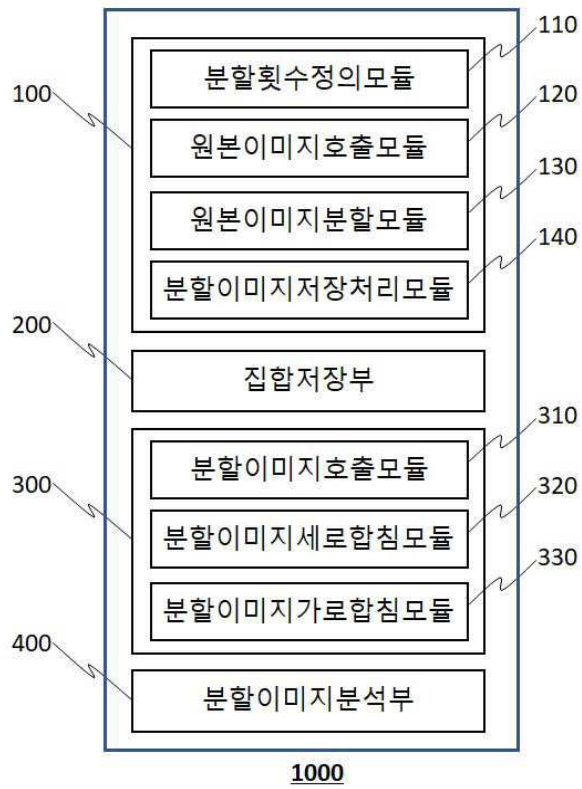
- [0101]
- [0102] 4ⁿ 분할 이미지를 통해 Data Augmentation을 수행하면, 기존 데이터만을 사용한 결과보다 개선된다는 것을 알 수 있다.
- [0103] 16분할의 경우, Baseline대비 0.6% 정도의 성능 향상을 이뤄냈으며, 64분할의 경우 보다 높은 약 0.8%가 향상된다.
- [0104] 이는 사용자에 따라 분할의 횟수를 증가시킬 수 있고, 더 많은 반복을 통해 데이터의 양을 증가시킴으로써 성능 향상을 이뤄낼 수 있음을 의미한다.
- [0105] 다만, 분할 횟수의 증가에 따라 동일 데이터 양 임에도 학습시간이 증가하고, FPS도 감소한다. 따라서, 적절한 분할 횟수를 설정하는 것이 중요할 수도 있을 것이다.
- [0106] 한편, 도 4의 경우, 4분할 이미지 생성 과정을 나타낸 방법이며, 도 5는 생성된 4분할 이미지를 나타낸 예시도이다.
- [0107] 그리고, 도 6의 경우, Data Augmentation 이전 시각화 이미지와 베이스 라인의 Data Augmentation 과정에서 Random Crop으로 인해 4 분할 이미지는 원본 이미지와의 차별점이 없어 실험에서 제외시켰다.
- [0108] 한편, 도 7의 경우, 16분할 이미지 생성 과정을 나타낸 방법이며, 도 8은 생성된 16분할 이미지와 16분할 이미지를 추가한 실험 결과값을 나타낸 예시도이다.
- [0109] 한편, 도 9의 경우, 64분할 이미지 생성 과정을 나타낸 방법이며, 도 10은 생성된 64분할 이미지와 64분할 이미지를 추가한 실험 결과값을 나타낸 예시도이다.
- [0110] 이때, 도 11에 도시한 바와 같이, 동일 설정, baseline과 64분할 이미지를 추가한 결과와의 비교 및 최종 시각화 결과를 나타낸 예시도이며, 3% 향상을 알 수 있었다.
- [0111] 본 발명을 통해, 기존의 데이터를 활용하여 새로운 데이터를 생성할 수 있으며, 분할 횟수에 따라 기존 데이터의 수보다 더 많은 데이터를 확보할 수 있다.
- [0112] 상기와 같은 내용의 본 발명이 속하는 기술분야의 당업자는 본 발명의 기술적 사상이나 필수적 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시된 것이며 한정적인 것이 아닌 것으로서 이해해야만 한다.

부호의 설명

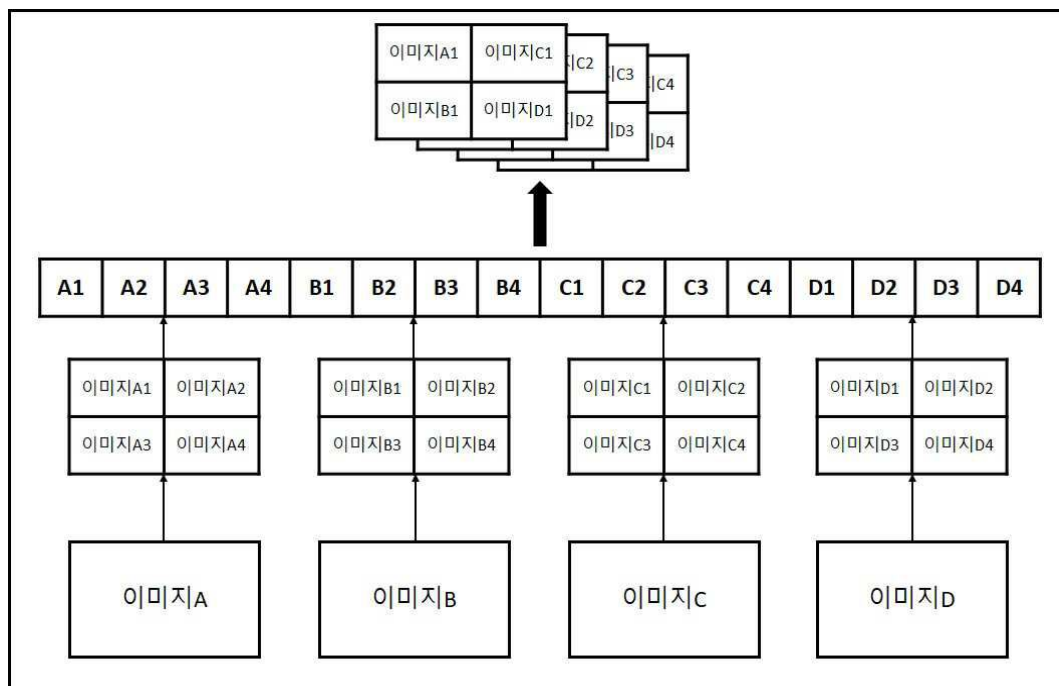
- [0113] 100 : 분할저장처리부
200 : 집합저장부
300 : 이어붙임처리부
400 : 분할이미지분석부

도면

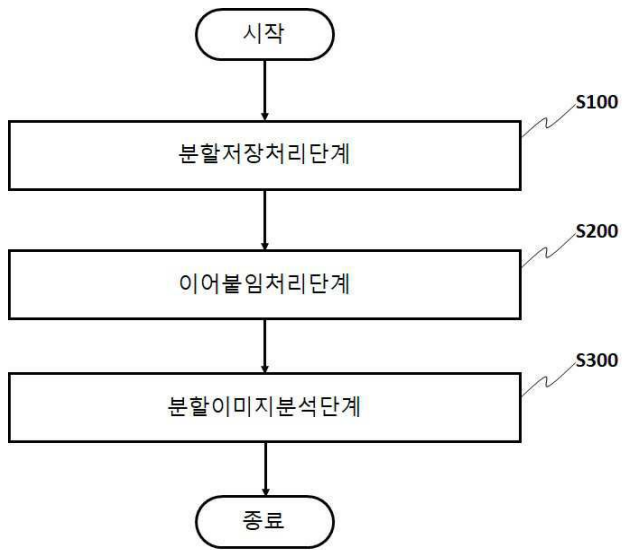
도면1



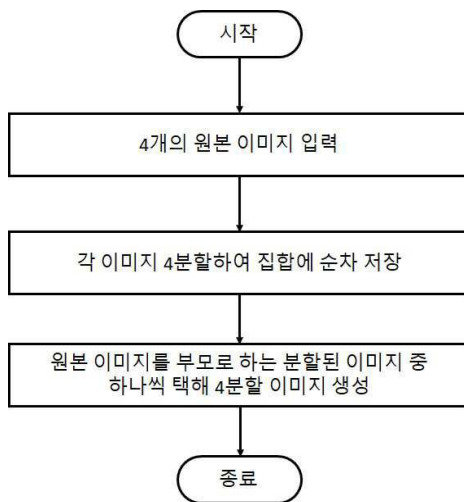
도면2



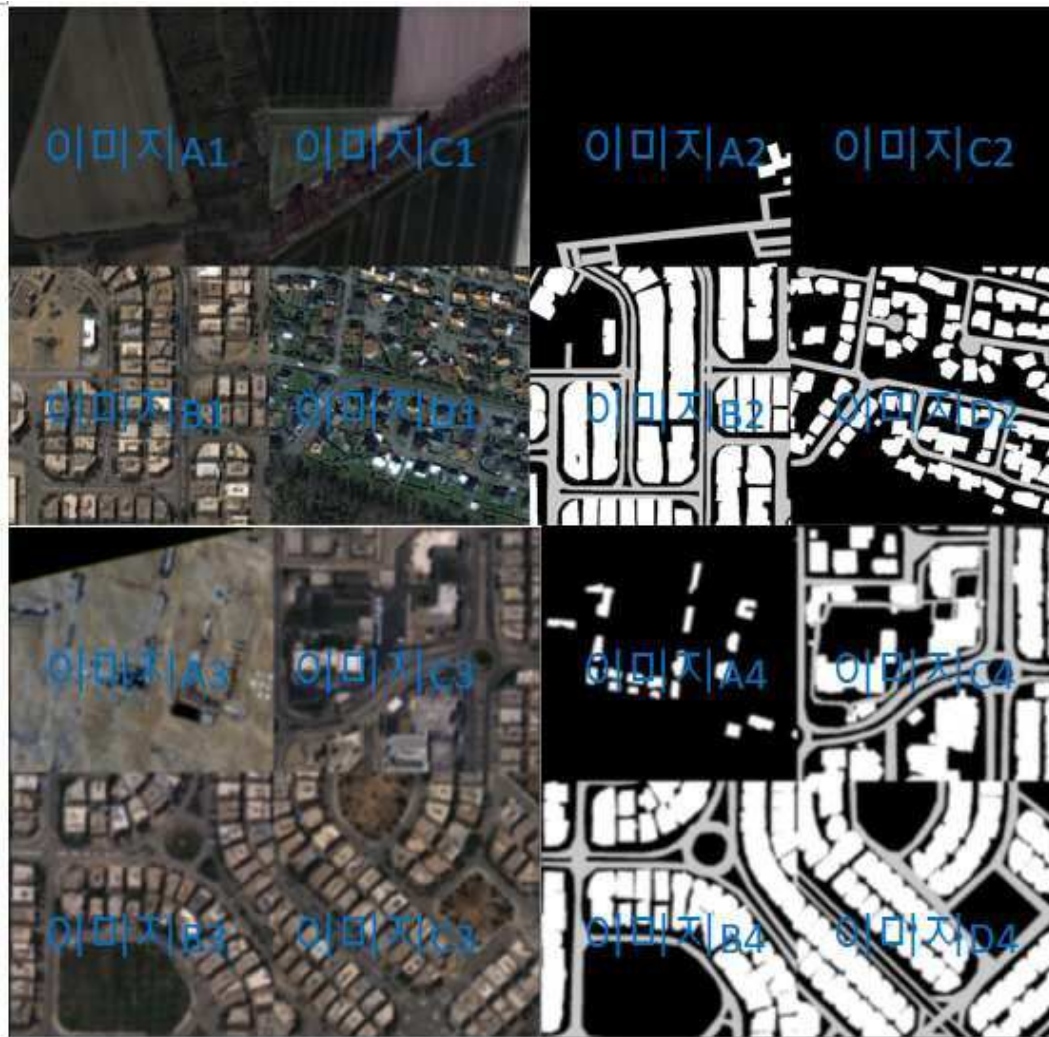
도면3



도면4



도면5



도면6

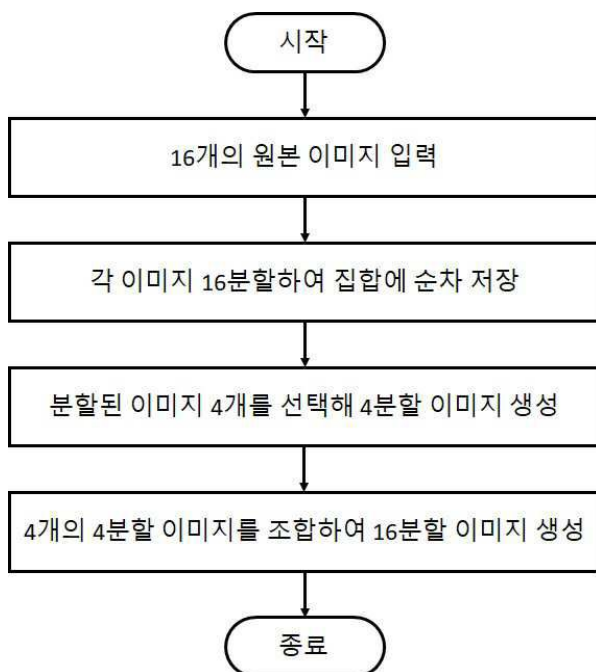


: Data Augmentation 이전 시각화



: baseline 의 Data Augmentation 과정에서 Random Crop 으로 인해 4분할 이미지는 원본 이미지와의 차별점이 없어 실험에서 제외된다.

도면7



도면8



: 생성된 16 분할 이미지

```
Epoch: 6
train: 100% ██████████ 134/134 [02:37<00:00, 1.18s/it, dice_loss = 0.226, iou_score = 0.6352]
valid: 100% ██████████ 142/142 [00:34<00:00, 4.07it/s, dice_loss = 0.1548, iou_score = 0.7437]
best : 0.7437055926297793

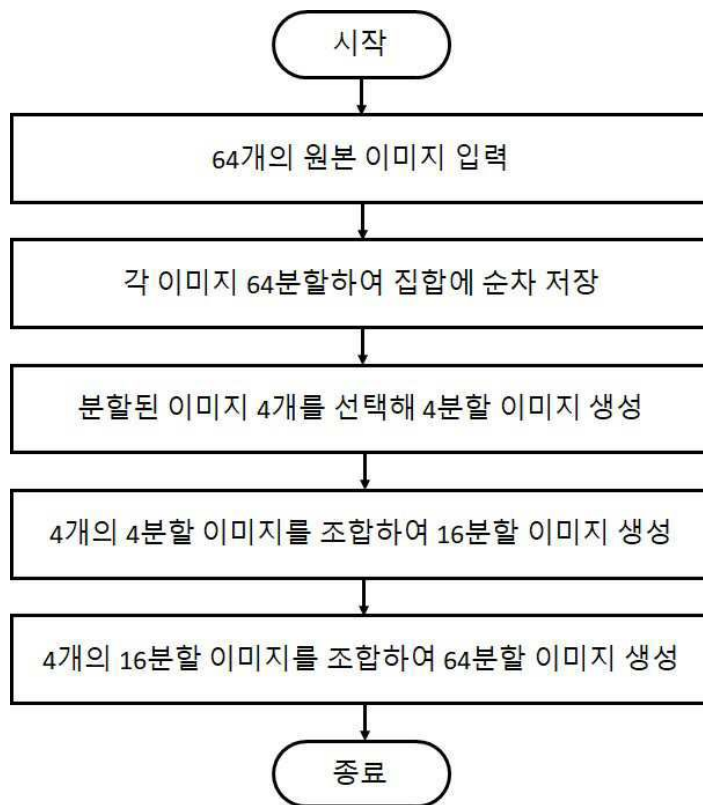
Epoch: 7
train: 100% ██████████ 134/134 [02:37<00:00, 1.18s/it, dice_loss = 0.2195, iou_score = 0.6434]
valid: 100% ██████████ 142/142 [00:34<00:00, 4.07it/s, dice_loss = 0.1625, iou_score = 0.7302]

Epoch: 8
train: 100% ██████████ 134/134 [02:37<00:00, 1.18s/it, dice_loss = 0.2211, iou_score = 0.6413]
valid: 100% ██████████ 142/142 [00:34<00:00, 4.07it/s, dice_loss = 0.1605, iou_score = 0.731]

Epoch: 9
train: 100% ██████████ 134/134 [02:37<00:00, 1.18s/it, dice_loss = 0.2149, iou_score = 0.6491]
valid: 100% ██████████ 142/142 [00:34<00:00, 4.07it/s, dice_loss = 0.143, iou_score = 0.7591]
best : 0.7590679765587123
max_score=0.7590679765587123
```

: 16 분할 이미지를 추가한 실험 결과

도면9



도면10



: 생성된 64 분할 이미지

```
Epoch: 7
train: 100% [████████████████████] 134/134 [04:44<00:00, 2.12s/it, dice_loss = 0.1977, iou_score = 0.6712]
valid: 100% [████████████████████] 142/142 [00:33<00:00, 4.20it/s, dice_loss = 0.131, iou_score = 0.7767]
best : 0.7767141771988126

Epoch: 8
train: 100% [████████████████████] 134/134 [04:44<00:00, 2.12s/it, dice_loss = 0.1984, iou_score = 0.6703]
valid: 100% [████████████████████] 142/142 [00:33<00:00, 4.20it/s, dice_loss = 0.1408, iou_score = 0.7624]

Epoch: 9
train: 100% [████████████████████] 134/134 [04:44<00:00, 2.12s/it, dice_loss = 0.1971, iou_score = 0.6724]
valid: 100% [████████████████████] 142/142 [00:33<00:00, 4.20it/s, dice_loss = 0.1318, iou_score = 0.7758]
max_score=0.7767141771988126
```

: 64 분할 이미지를 추가한 실험 결과

도면11

```
Epoch: 97
train: 100% ██████████ 53/53 [00:26<00:00, 2.00it/s, dice_loss = 0.1742, iou_score = 0.7142]
valid: 100% ██████████ 36/36 [00:32<00:00, 1.11it/s, dice_loss = 0.1106, iou_score = 0.8057]
0.8069081207116444

Epoch: 98
train: 100% ██████████ 53/53 [00:26<00:00, 2.00it/s, dice_loss = 0.1599, iou_score = 0.7333]
valid: 100% ██████████ 36/36 [00:32<00:00, 1.11it/s, dice_loss = 0.1106, iou_score = 0.8054]
0.8069081207116444

Epoch: 99
train: 100% ██████████ 53/53 [00:26<00:00, 2.02it/s, dice_loss = 0.1537, iou_score = 0.7432]
valid: 100% ██████████ 36/36 [00:32<00:00, 1.11it/s, dice_loss = 0.1104, iou_score = 0.8057]
0.8069081207116444
max_score=0.8069081207116444

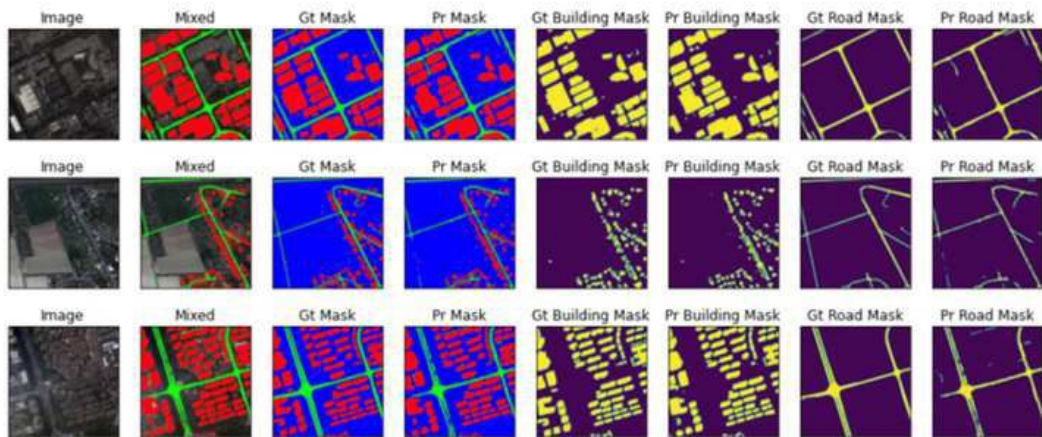
Epochs: 93
train: 100% ██████████ 267/267 [04:54<00:00, 1.10s/it, dice_loss = 0.1318, iou_score = 0.7698]
valid: 100% ██████████ 142/142 [00:32<00:00, 4.39it/s, dice_loss = 0.09437, iou_score = 0.834]
Model saved!
0.8339658938872986

Epochs: 97
train: 100% ██████████ 267/267 [04:55<00:00, 1.11s/it, dice_loss = 0.1352, iou_score = 0.7644]
valid: 100% ██████████ 142/142 [00:32<00:00, 4.38it/s, dice_loss = 0.09485, iou_score = 0.8332]
0.8339658938872986

Epochs: 98
train: 100% ██████████ 267/267 [04:55<00:00, 1.11s/it, dice_loss = 0.136, iou_score = 0.763]
valid: 100% ██████████ 142/142 [00:32<00:00, 4.38it/s, dice_loss = 0.09568, iou_score = 0.8317]
0.8339658938872986

Epochs: 99
train: 100% ██████████ 267/267 [04:55<00:00, 1.11s/it, dice_loss = 0.132, iou_score = 0.7691]
valid: 100% ██████████ 142/142 [00:32<00:00, 4.38it/s, dice_loss = 0.09517, iou_score = 0.8326]
0.8339658938872986
max_score=0.8339658938872986
```

: 동일 설정, baseline 과 64 분할 이미지를 추가한 결과와의 비교(3% 향상)⁴⁾



: 최종 시각화 결과⁴⁾