KGU 경기대학교

영상기반 음영지역 보정 시제품 제작 및 시험

2019. 05. 24.

Byoung-Dai Lee

Division of Computer Science & Engineering

Kyonggi University

C/O/N/T/E/N/T/S

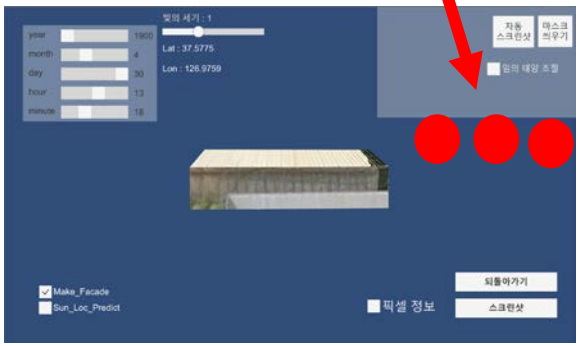
- 그림자 이미지 생성 모듈 보완
- 딥러닝 기반 그림자 제거 알고리즘 개발

Facade 기능 개발 (1)

- 시나리오

[건물 선택]

회전 버튼



저장

[회전을 통한 정면 이미지]

[주변 건물 제거 및
확대 후 카메라 정면 배치]

Facade 기능 개발 (2)

▪ 개발 완료 내용

- 클릭 시 주변 건물 제거, 텍스쳐 원본 유지 (광원 효과 없음)

[Make Façade 기능 순서]

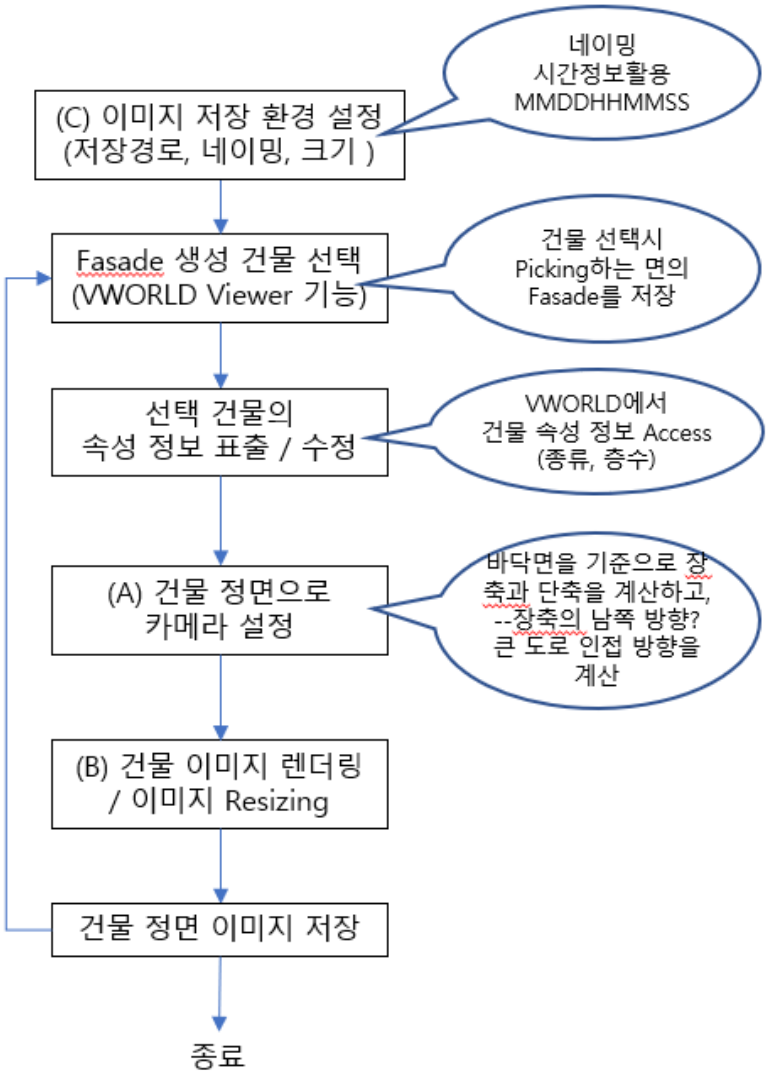
1. **"Make_Façade" 토글 클릭**
2. **건물 선택**
3. **정면 이미지 캡처 (개발 중)**

Facade 기능 개발 (3)

- **현재 개발 중**
  - 정면을 바라볼 수 있도록 건물 회전 방법 구현 중

Facade 기능 개발 (4)

- 문의 사항
  - (A) 바닥면을 선택할 수 있는지? 도로 인접 방향은 불가능해 보임
  - 건물 선택 시 Picking하는 면은 사실상 어려워보임
  - 가로 세로 기준의 애매모호함



태양 위치 추적 기능 (1)

- 시나리오

바닥에 그림자 있는 건물

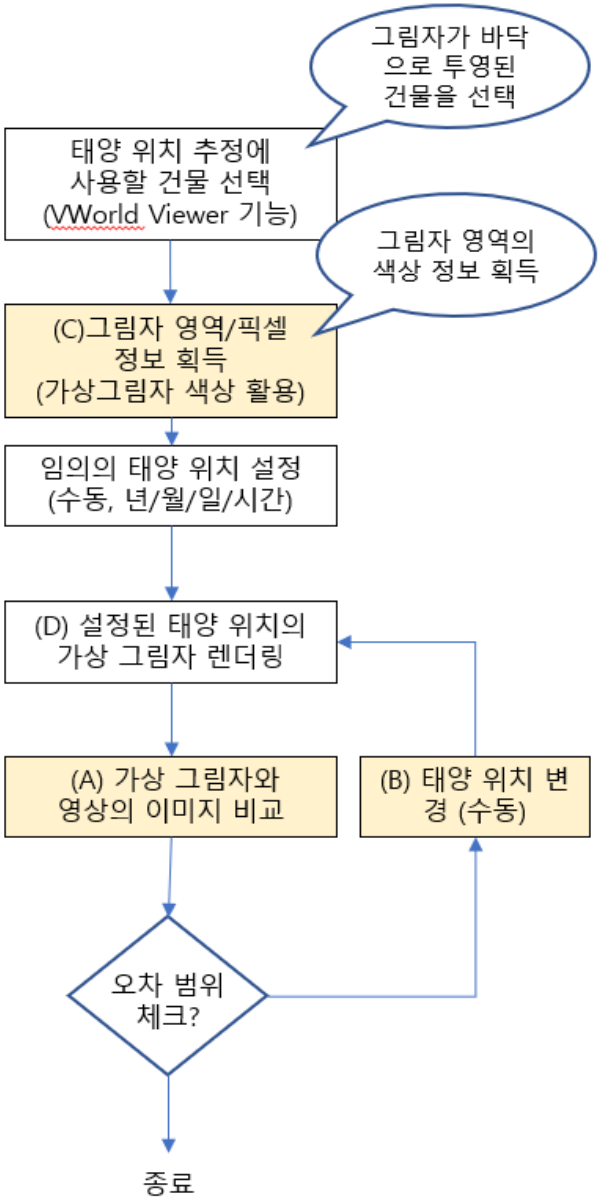
수동 조절하여 태양 위치 추정

더블 클릭으로 접근

태양 위치 추적 기능 (2)

▪ 문의 사항

- Vworld Viewer 기능이 무엇인지? (Unity? Vworld?)
- 그림자 영역/픽셀 정보 획득? 가상 그림자 색상 활용?
- 설정된 태양 위치의 가상 그림자 렌더링이 기존 Unity속 태양이 맞는지?
- 가상 그림자와 영상의 이미지 비교를 통한 오차범위는 눈으로 하는 것인지?
- 태양의 위치 설정 공식?
- 그림자 영역 선택은 불가능해 보임. 선택 지점의 픽셀 정보는 획득 가능



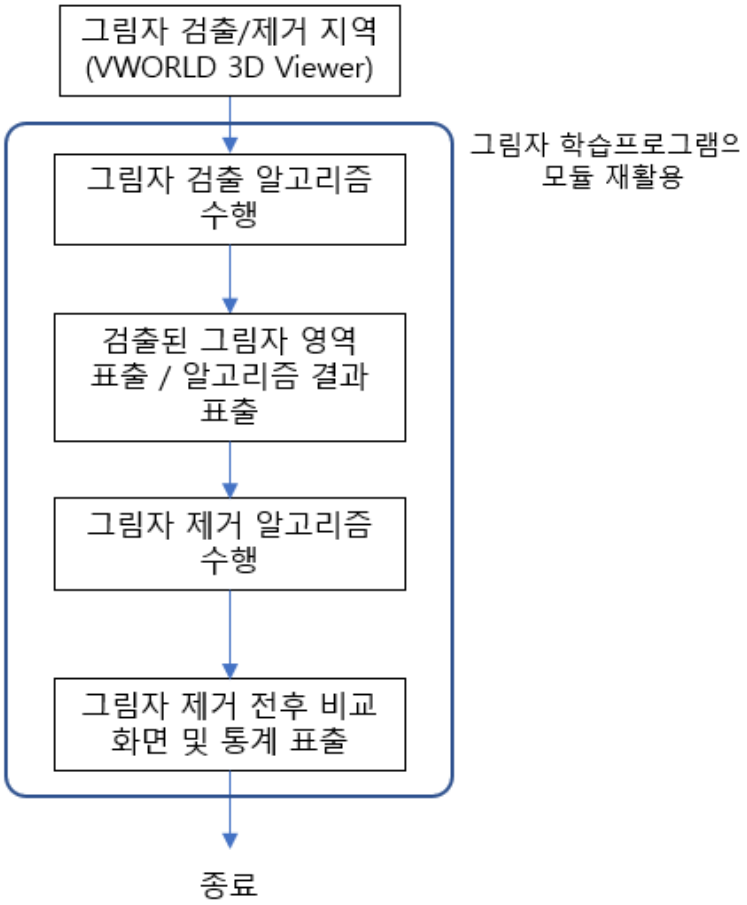
그림자 검출 및 제거 모듈 (1)

- 시나리오
  1. 현재 화면 캡처 이미지 또는 저장된 이미지 선택
  2. Python 코드를 수행하는 명령어 입력으로 그림자 학습 프로그램의 모듈 실행
  3. 모듈내부에서 그림자영역 표출 및 알고리즘 결과 표출 등의 출력 저장
  4. 모듈 내부에서 제거 알고리즘 수행
  5. 모듈 내부에서 그림자 제거 전후 비교 화면 및 통계 표출 저장

그림자 검출 및 제거 모듈 (2)

▪ 문의 사항

- 건물, 지형, 도로 레이어 선택 기능?
- 학습 모델을 활용한 그림자 검출 및 그림자 표출 On/OFF는 프로그램 내부에서 표현하기 힘듦
- 객체별 텍스처와 그림자 중첩부분 추출과 저장은 Unity에서 안 해도 될 수 있음

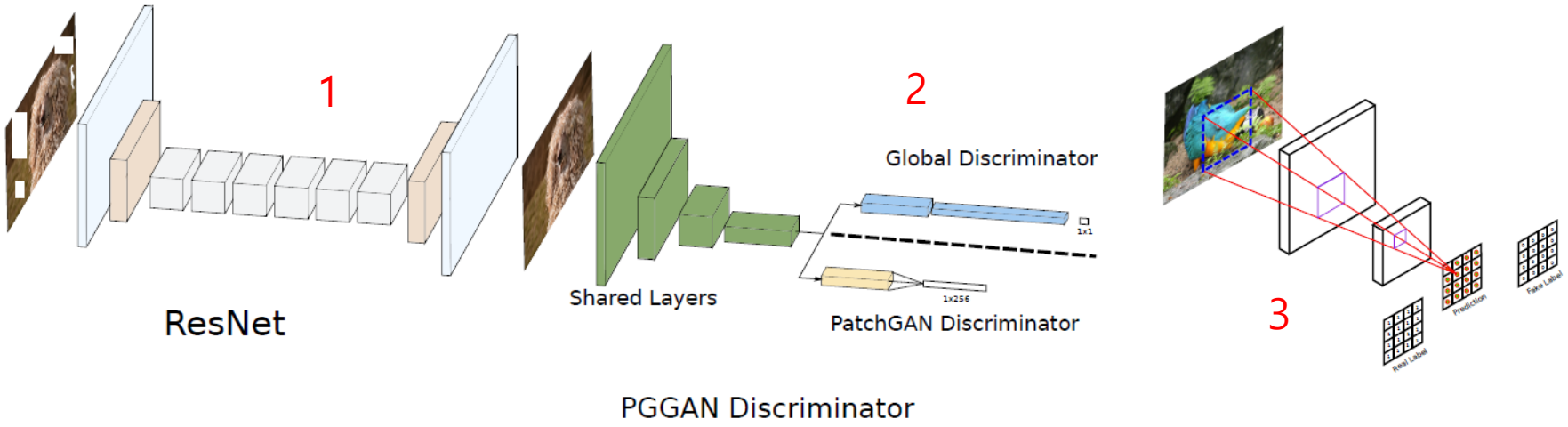


딥러닝 기반 그림자 제거 (1)

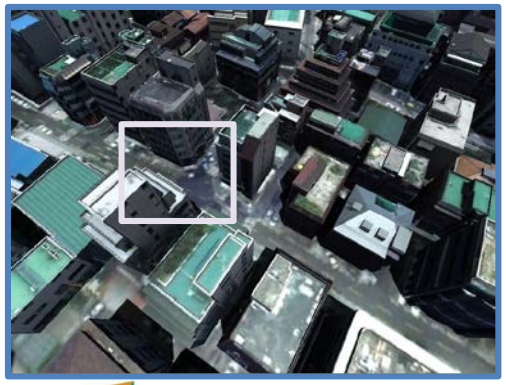
- **그림자 제거를 위한 딥러닝 기반 코드 개발**
  - Discriminator
    - Patch GAN에서 제시된 idea 적용
  - Generator
    - Residual Block의 다양화
    - Dilated Convolution 적용
  - Loss 수정
    - Joint loss 사용

딥러닝 기반 그림자 제거 (2)

- Patch-based Image Inpainting with GAN (2018)



2. Discriminator

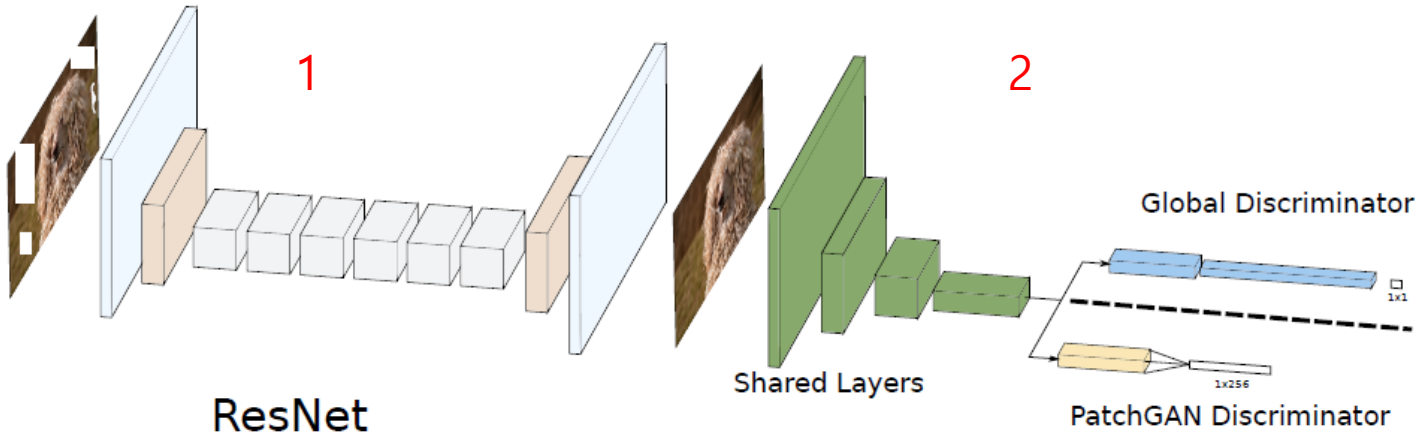


Discriminator 부분의 개념

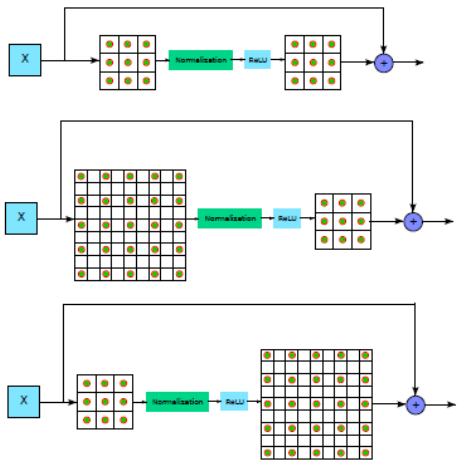
- (Patch + Global) => PGGAN Discriminator
  패치 단위와 전체 이미지 단위로 나누어 진짜 or 가짜 인지 판단하기 때문에, 패치 영역으로 좀 더 강조하여 판단할 수 있는 효과가 기대됨.
- 기존의 전체 이미지 단위로만 판단한 것을 보완

딥러닝 기반 그림자 제거 (3)

- Discriminator

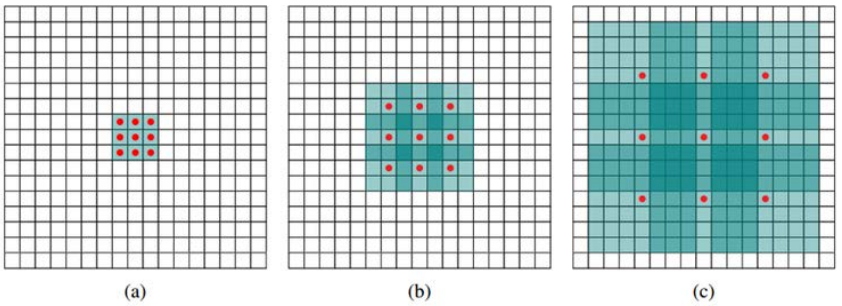


1. ResNet block

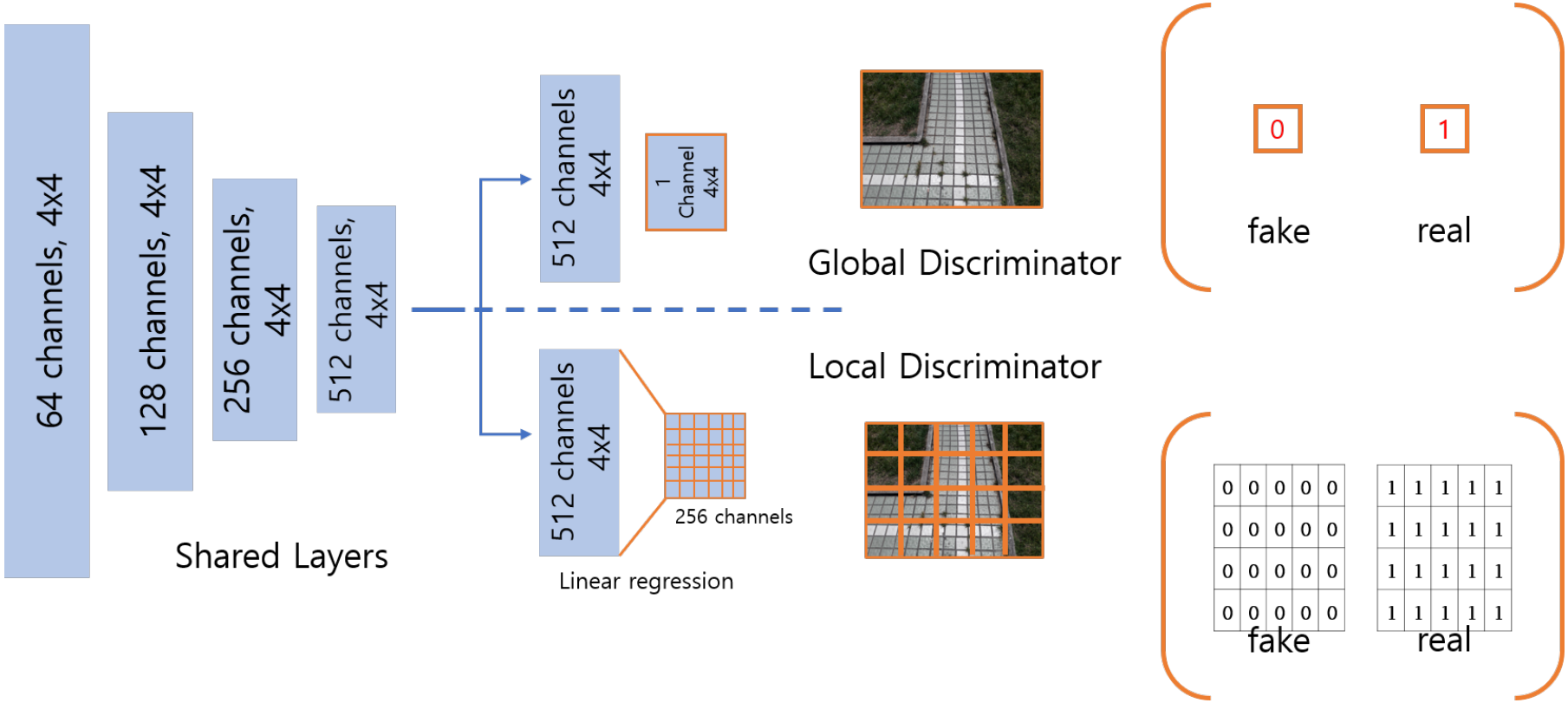


PGGAN Discriminator

3

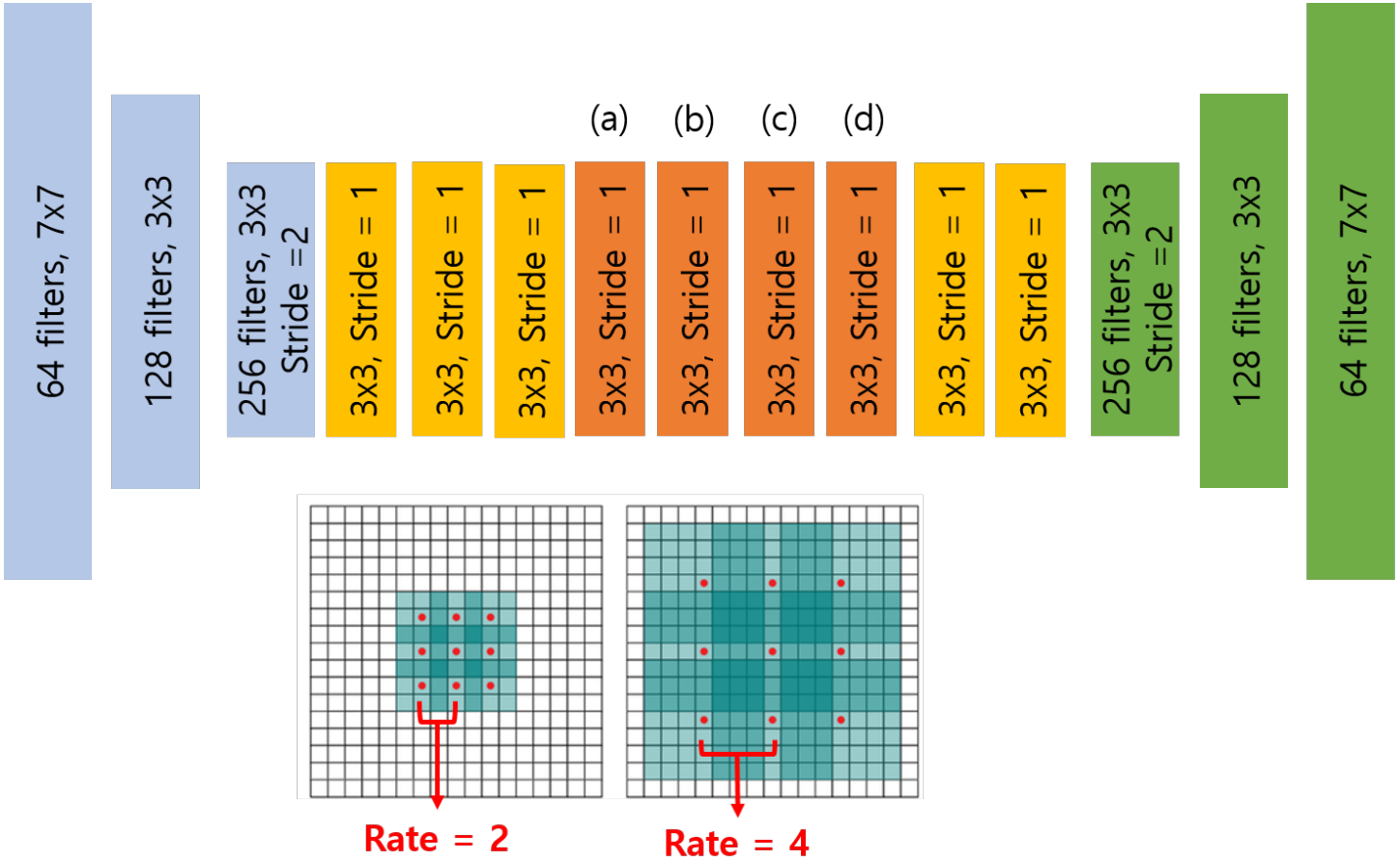


딥러닝 기반 그림자 제거 (4)



딥러닝 기반 그림자 제거 (5)

- Generator



(a) Dilation rate =2
(b) Dilation rate = 4
(c) Dilation rate = 8
(d) Dilation rate = 16

Dilated Residual Block
Residual Block
deconvolution
convolution

딥러닝 기반 그림자 제거 (6)

Joint Loss Function

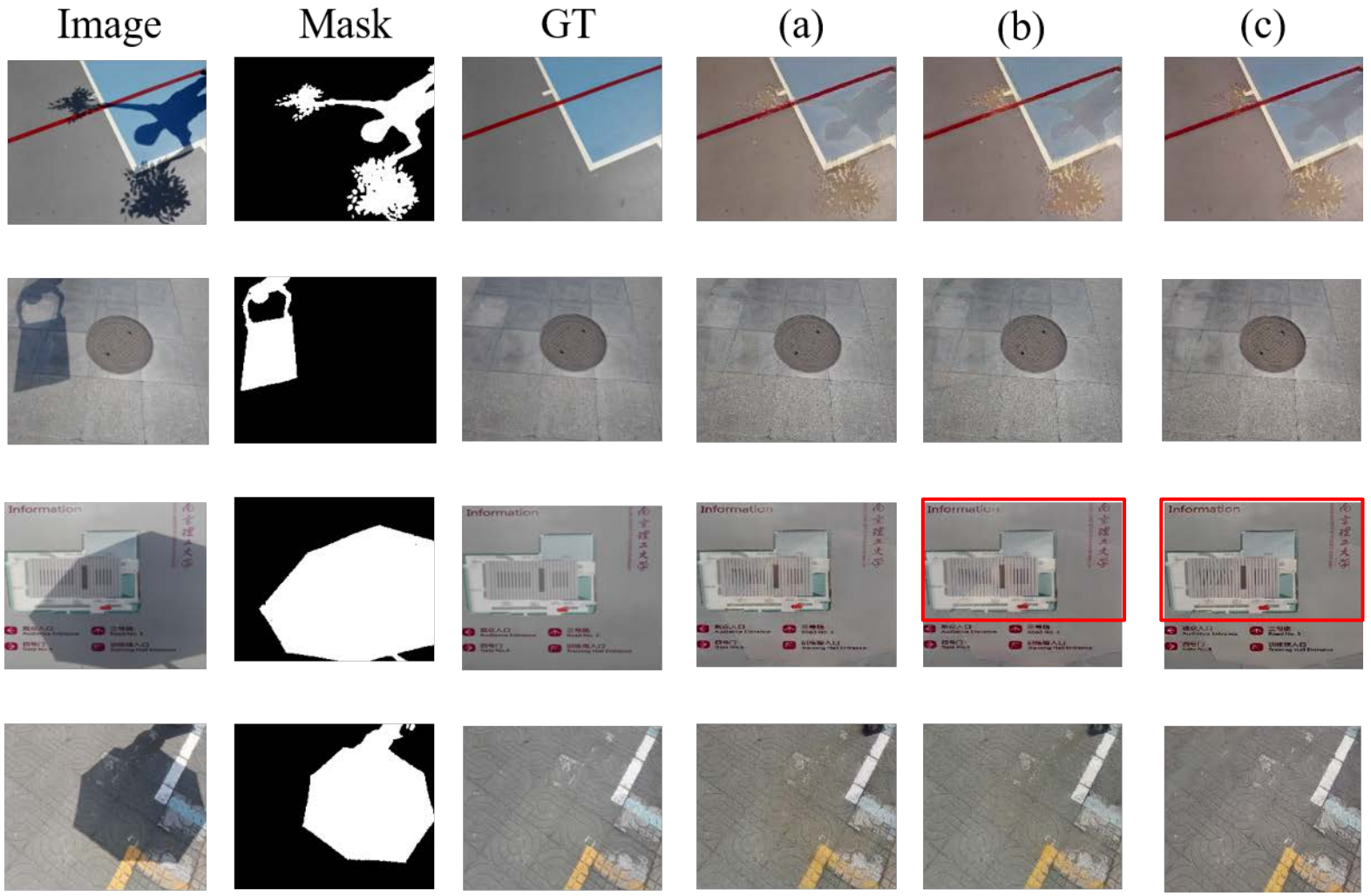
- Reconstruction loss : 발생기에서 만든 이미지(가짜)와 GT 이미지 (진짜)의 픽셀단위 L1 distance 비교
- Adversarial loss : 분류기에서 G와 D를 동시에 학습
- Joint loss : 각 구성요소들을 비율에 따라 하나의 식으로 구성

  λ_1 = 0.995, $\lambda_2 = 0.0025$, $\lambda_3 = 0.0025$

  - $L_{rec} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{WHC}\|y - x\|$
  - $L_{GAN}(G, D) = E_{x\sim p(x)}[logD(x)] + E_{y\sim pG(\tilde{x})}[\log(1 - D(G(\tilde{x})))]$
  - $L = \lambda_1 L_{rec} + \lambda_2 L_{g\_adv} + \lambda_3 L_{p\_adv}$

딥러닝 기반 그림자 제거 (7)

- 실험 결과



딥러닝 기반 그림자 제거 (6)

- 큰 차이는 없지만 미세한 부분들이 좀 더 자세히 복원

(b)

(c)

딥러닝 기반 그림자 제거 (7)

▪ 성능 평가 방법

- RMSE(Root Mean Squared Error)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - y_i)^2}$$

$p_i =$ 생성된 이미지

$y_i =$ 정답 이미지

방법	성능 값(RMSE)
기존 데이터 셋(ISTD)	10.5153
생성기(Residual Block 6)	5.0789
생성기(Residual Block 9)	4.9220
Patch-based image	5.0883

딥러닝 기반 그림자 제거 (8)

▪ 향후 진행 방향

- 성능 평가 방법 개선
  - 그림자 영역과 비그림자 영역을 나누어 진행
- 네트워크
  - Dilated residual block의 조합을 다양화하여 테스트 및 평가
  - Attention block[1]을 활용한 이미지 복원 방법 적용

[1] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. arXiv preprint arXiv:1801.07892 (2018)

Question or Comments?