

****학회 [2020 추계학술대회]**

건물 외피 영상에서의 oooooo 탐지 및 oooooo oooooo 연구

목차

01 연구 개발 개요

- 1.1 연구 개발 배경
- 1.2 선행 연구
- 1.3 연구 개발 필요성
- 1.4 연구 개발 계획

02 연구 개발 프로세스

- 2.1 Dataset Construction
- 2.2 Object Detection
- 2.3 Edge Detection
- 2.4 Segmentation
- 2.5 Simulation

03 연구 결과

- 3.1 연구 결과 최종 모델
- 3.2 detection 결과
- 3.3 정량 지표 결과
- 3.4 시뮬레이션 이미지 데이터 구축 결과

04 연구 결론

- 4.1 한계점
- 4.2 향후 연구 발전 방향

01 연구 개발 개요

1. 연구개발 개요

1-1. 연구 개발 배경

외피 영상 정보로부터 부위별 위치탐지(Localization) 및 의미론적 분할(Semantic segmentation)의 연구를 통해 외피 구성 정보를 분석하고 이의 활용 방안을 고찰함

- 연구를 위해서는 컴퓨터 비전 관련 기술과 이미지 라벨링 기술, 기계학습, 합성곱 신경망, 통계처리 기술 연구가 필요함
- 기존 건물에 대한 분석 시, 외피 구성정보 파악 및 국소부위를 추출할 수 있는 기술이 필요
 - ▶ 건물 외벽으로부터 추출한 이미지를 통해 Image detection 및 Segmentation모델 개발이 필요

1. 연구개발 개요

1-2. 선행 연구

<국내동향>

- 김정문. *컨볼루션 신경망을 이용한 사이드 스캔 소나 이미지 인식 및 수중물체 탐지에 관한 연구*. Diss. 한양대학교, 2018.
 - ▶ 문서 및 이미지를 디지털화 하는 OCR(Optical Character Recognition)기법은 특정내용에 대한 내용만을 탐색하지 못하는 한계점이 존재, 이를 해결하기위해 Faster-RCNN 을 이용하여 특정내용에 대한 탐색 및 정보추출을 수행
- 강지수, et al. "교통 영상 빅데이터 처리를 위한 Yolo 기반 광원 객체 탐지." *융합정보논문지 (구 중소기업융합학회논문지)* 10.8: 40-46.
 - ▶ 교통 영상 빅데이터에서 객체 인식 및 탐지에 대한 연구. YOLO기반으로 타 연구에서 제한적으로 발생하는 야간 도로에서의 객체인식을 위해 색상 모델 변화를 적용하여 인식을 보완

<국외동향>

- Ali, Haider, et al. "Window detection in facades." *14th International Conference on Image Analysis and Processing (ICIAP 2007)*. IEEE, 2007
 - ▶ façade Data를 활용하여 도시환경에서의 창문을 탐지해내는 프로세스를 진행. 단일 창문을 탐지해내는데 효과적이었으며 모바일 서비스 및 3D 모델링에 활용될 가능성이 있음
- Wang, Ruisheng, Jeff Bach, and Frank P. Ferrie. "Window detection from mobile LiDAR data." *2011 IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 2011.
 - ▶ LiDAR(Light Detection And Ranging)데이터를 통해 도시에서의 창문 탐지를 진행

1. 연구개발 개요

1-3. 연구 개발 필요성

연구 목적

건물 외피 영상에서의 외피 부위별 위치탐지 및
의미론적 분할 연구

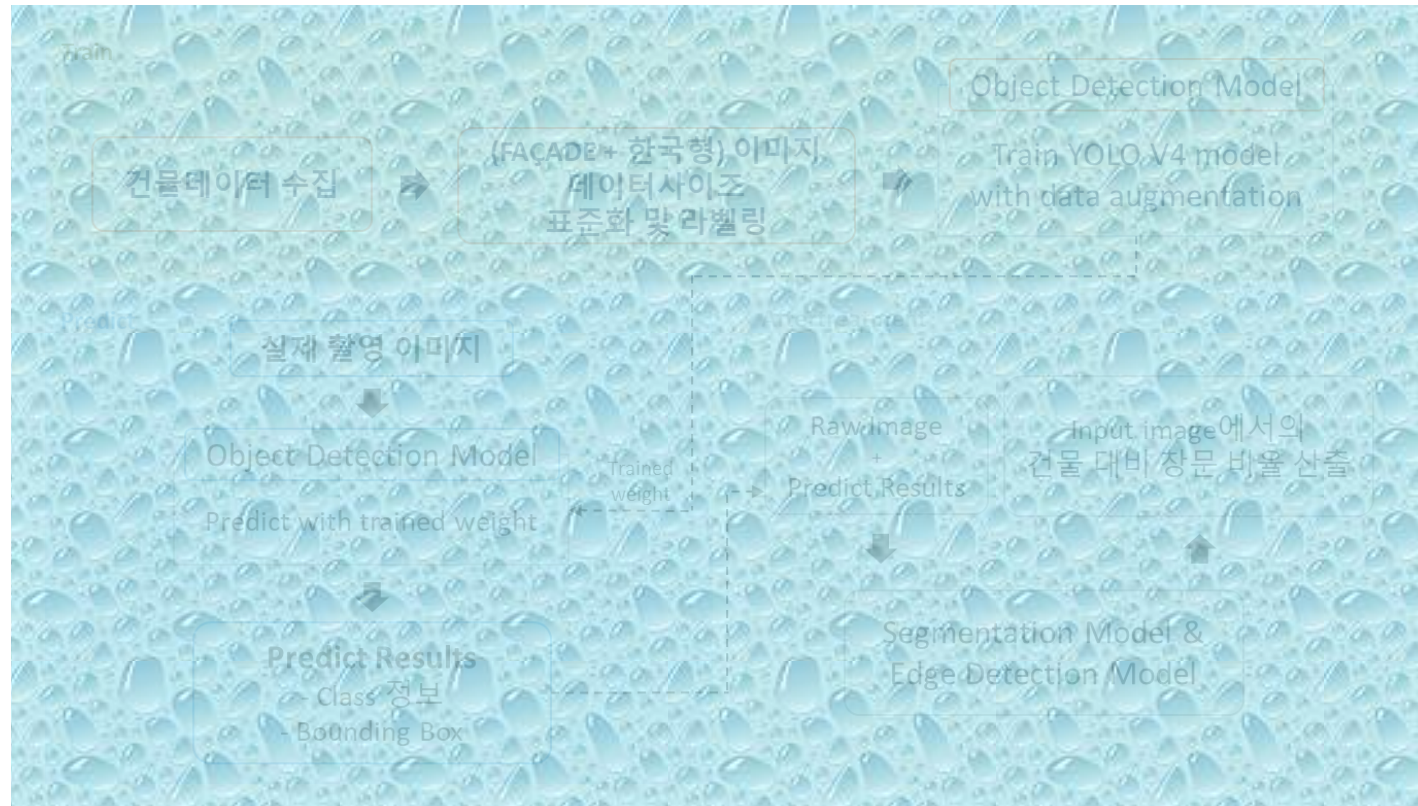
- 건축물 이미지 분석을 위한 특징 추출
프로세스 개발

연구 범위

- 건축물 외피 이미지 및 이미지 라벨 샘플 DB구축

1. 연구개발 개요

1-4. 연구 개발 계획



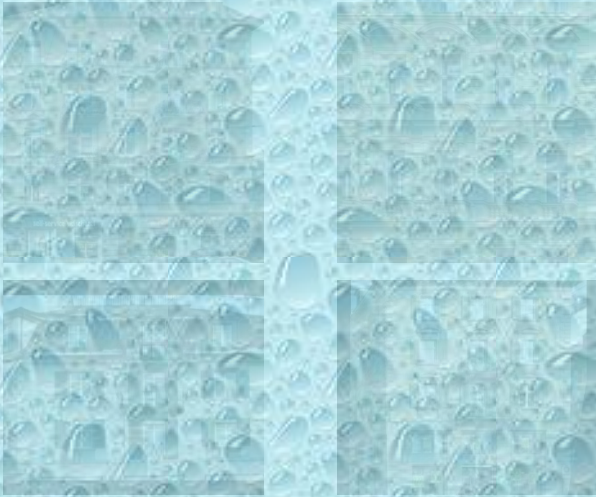
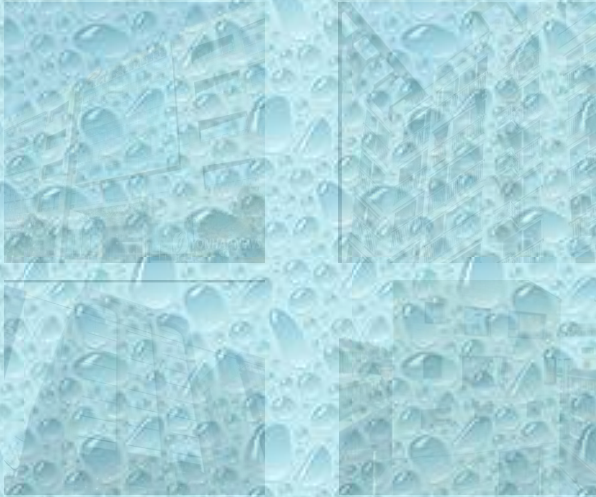
02 연구 개발 프로세스

2. 연구개발 프로세스

2-1. Dataset Construction

- 건물 외피 영상에서의 외피 부위별 위치 탐지를 위한 건물 외벽 이미지 데이터 구축

- 유럽형 건물 facade image data 374장, 한국형 건물 image data 567장으로 총 841장의 dataset 구축
- 유럽형 건물 Facade image data 및 한국형 건물 image data의 특징

Fasade image	한국형 image
<ul style="list-style-type: none">- 저층 건물 위주로 image 구성- 유사한 형태의 건물구조와 창문이 많음- 건물 Image의 구도가 정면에서 표현된 Data가 다수	<ul style="list-style-type: none">- 상가, 아파트, 빌딩 등 다양한 건물 형태와 창문 존재- 통유리를 외벽으로 사용하거나 통유리 사용 면적이 넓은 건물이 존재
	

2. 연구개발 프로세스

2-1. Dataset Construction

- 데이터 구축 방법 (labeling tool)

- Object Detection은 image 내 포함 된 객체(object)의 class와 위치를 지정하기 위한 labeling 작업이 필요
- Image labeling용 tool(Labelimg)을 이용하여 YOLO 모델 학습에 적합한 labeling format 구축
- YOLO v4용 학습 데이터를 위한 bounding box labeling 진행
 - label 정보 : { class(number) ' $\frac{x}{coord}$ ' $\frac{y}{coord}$ ' $\frac{w}{width}$ ' $\frac{h}{height}$ }
- Image file별 annotation file 생성

YOLO 버전 파일 생성

이미지 내 생성된 Class

Bounding Box 그리기

Class 지정

Labelimg tool을 이용해 labeling 진행

<image file>

<annotation file>

img254.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움(H)

```
0 0.736133 0.643555 0.134766 0.130859
0 0.279297 0.645508 0.117188 0.119141
0 0.429688 0.361328 0.121094 0.121094
0 0.789062 0.351562 0.121094 0.125000
0 0.427734 0.049805 0.113281 0.005703
10 0.609375 0.942383 0.074219 0.111328
0 0.381836 0.943359 0.232422 0.109375
```

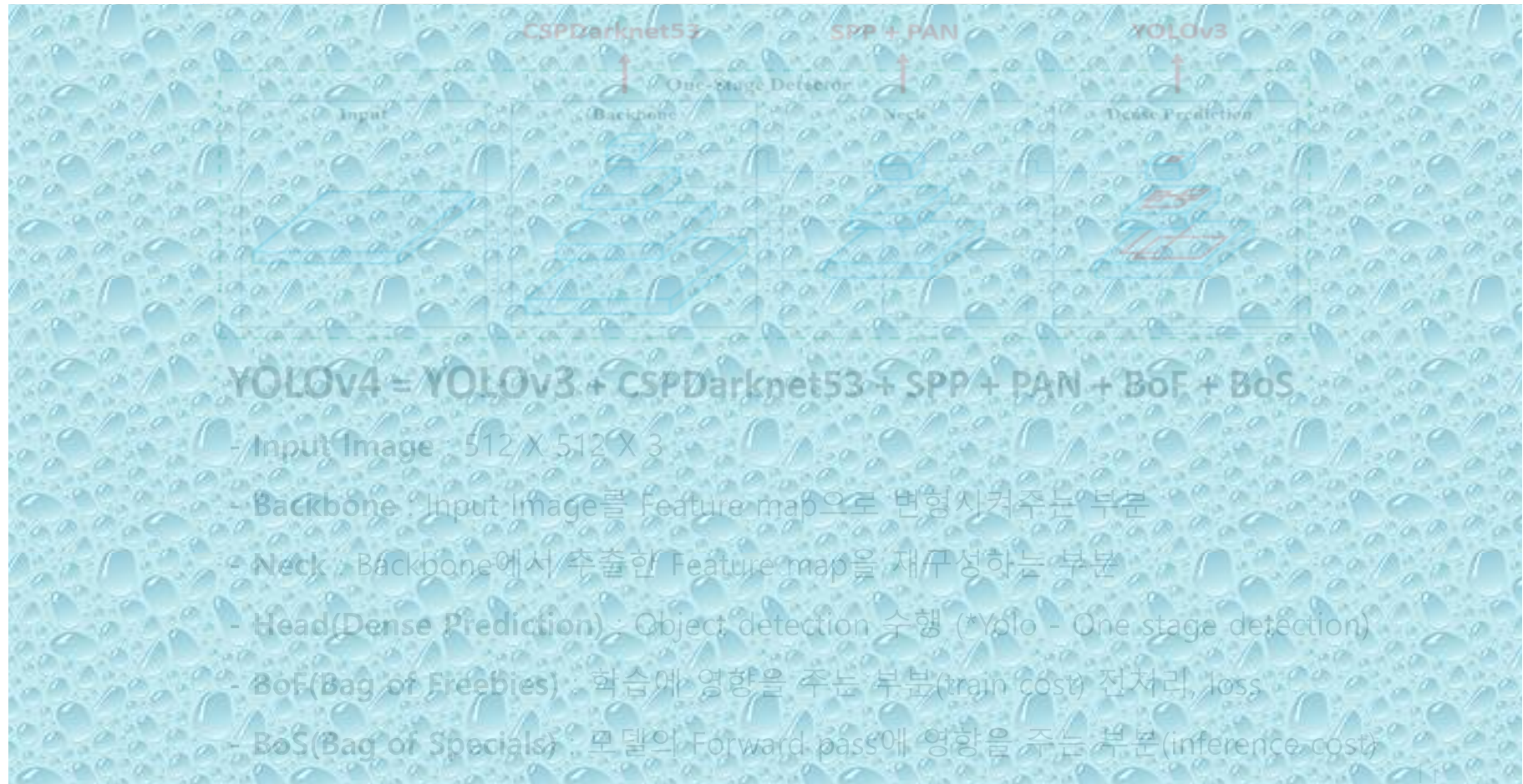
class x y w h

0 0.279297 0.645508 0.117188 0.119141

2. 연구개발 프로세스

2-2. object detection

- 객체(object)의 종류(class)와 그 위치(bounding box)를 특징하는 문제
- 본 연구에서는 YOLO v4 model을 baseline model로 선정



2. 연구개발 프로세스

2-3. edge detection

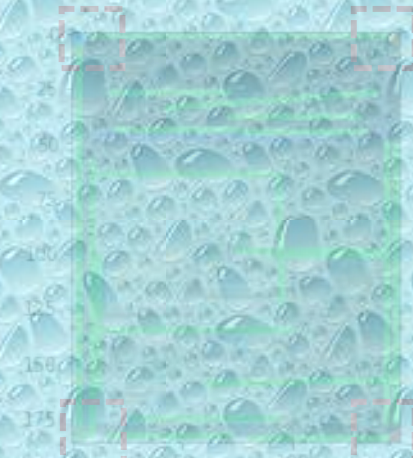
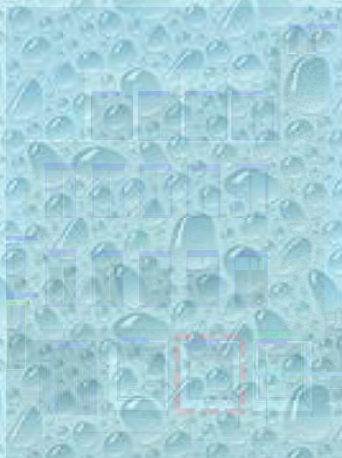
- Python OpenCV의 Canny Edge Detection을 이용
 - input image(cropped window)의 모든 edge detection 진행

- 창문의 면적 계산을 위한 좌표 검출

- crop된 image의 모서리 부분과 가장 가까운 좌표를 찾아 창문 면적 비율의 계산에 이용
- 바르게 crop 되었다면 가장자리와 가장 가까운 좌표가 창문의 좌표로 유추할 수 있음

- Cropped image 각각에서 창문의 면적 계산

- threshold을 정하여 detecting이 잘못 된 창문들은 제외한/후에 면적 비율 계산 진행
- 하나의 image에서 detection되어 crop된 모든 창문의 면적 비율을 유추된 좌표를 통하여 계산



2. 연구개발 프로세스

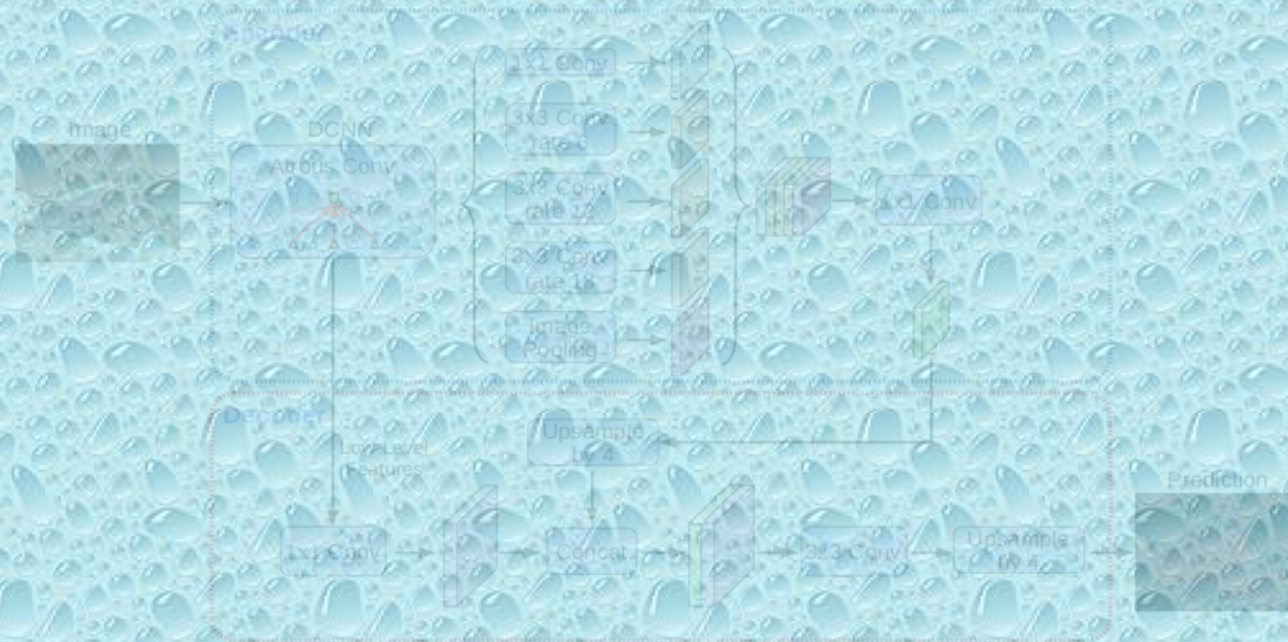
2-4. segmentation

- 실험 목적

- 건물의 탐지 및 건물 외벽의 창문 탐지
- 건물과 창문의 면적 산정을 통한 비율 산출
 - > 일반적인 카메라로 촬영된 이미지로 건물과 건물 외벽의 객체를 최종적으로 분류하기 위한 연구

- 실험 방법 : Deeplab v3+ 활용

- 동일한 class의 instance는 구분하지 않는 Semantic segmentation 모델 사용



< Deeplab v3+ architecture >

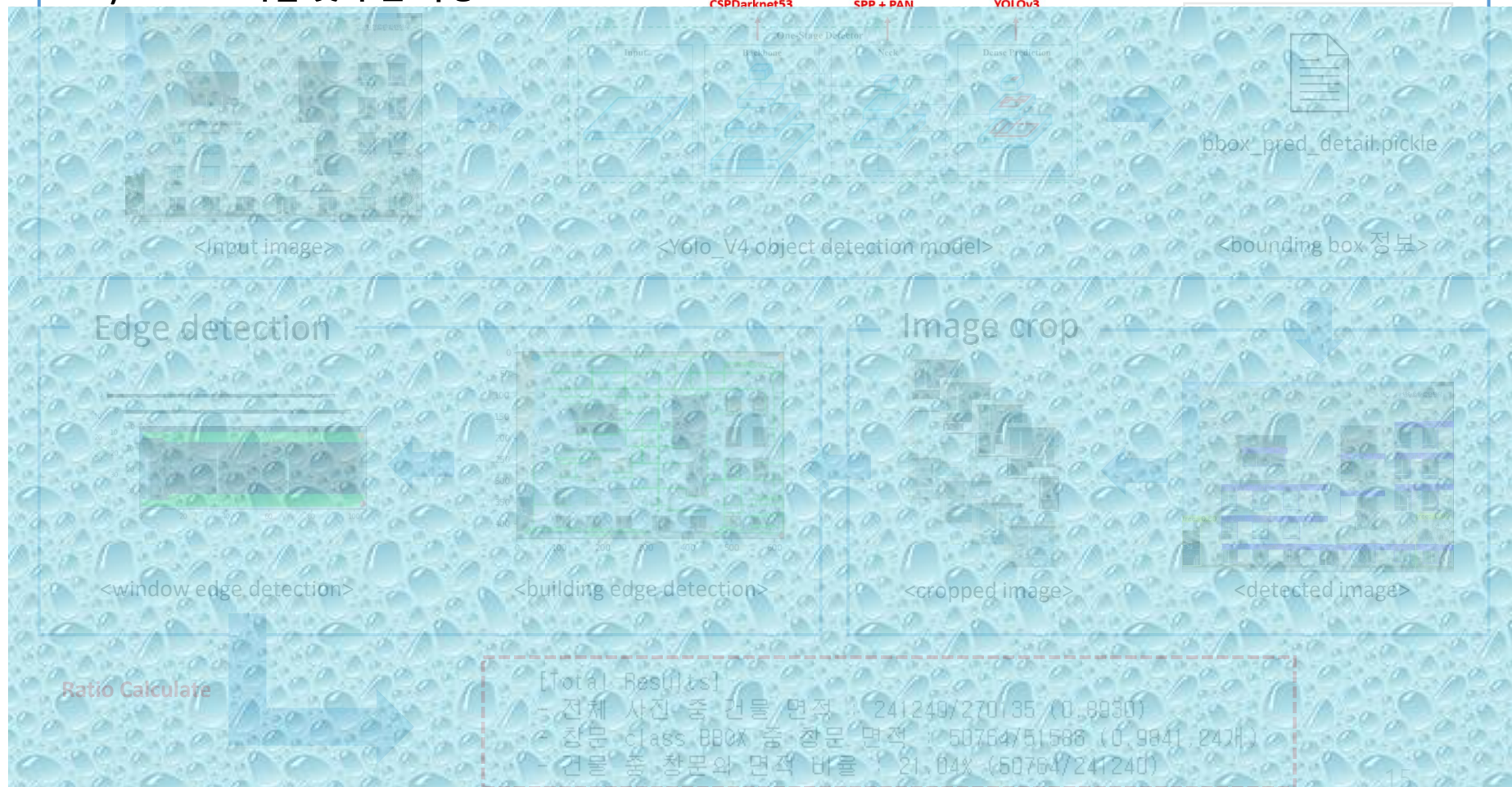
03 연구 결과

3. 연구 결과

3-1. 최종 모델 설명

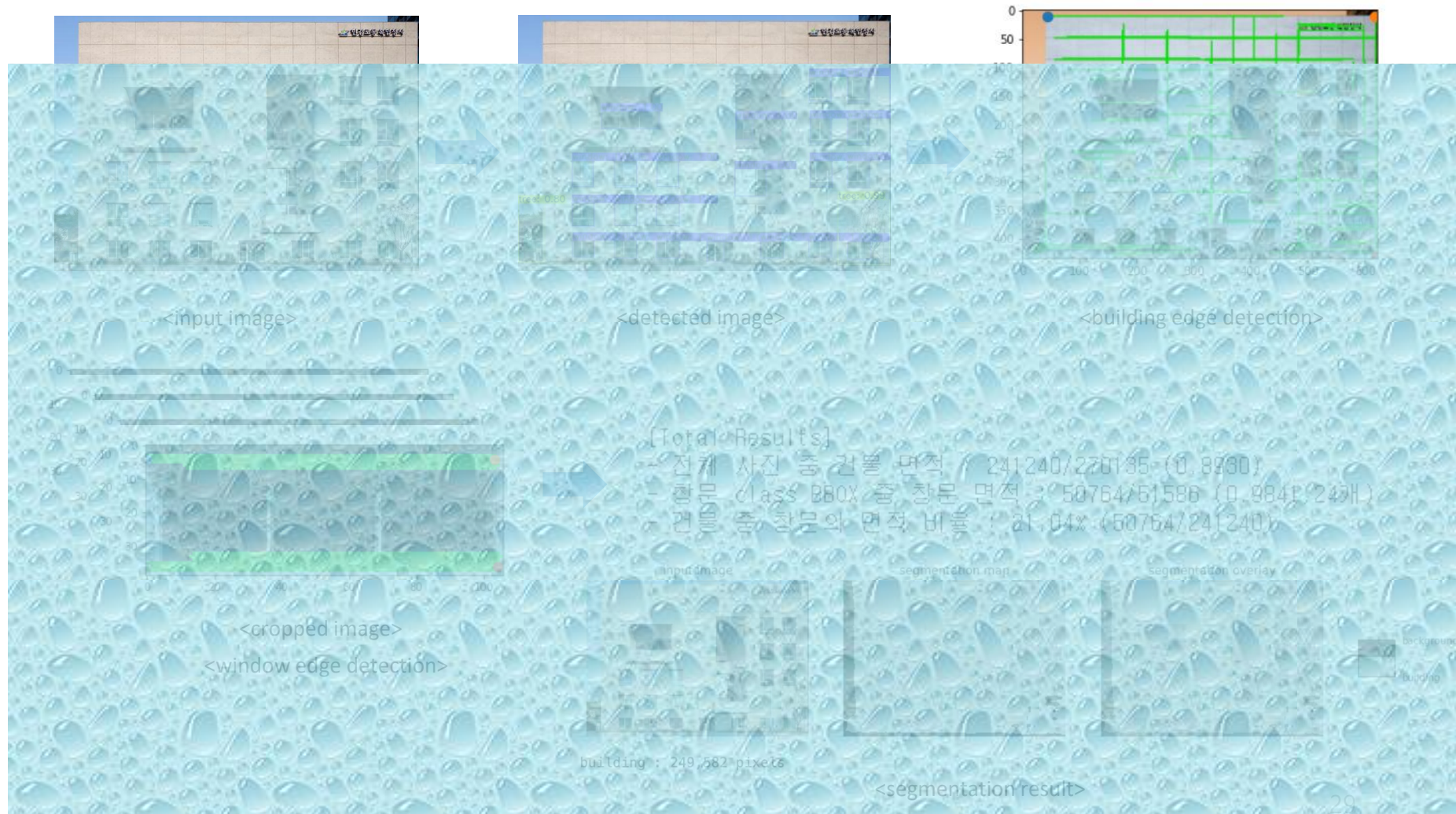
Window detection

1) YOLO V4 학습 및 추론 과정



3. 연구 결과

3-2. detect 결과 (정면 사진)



3. 연구 결과

3-3. 성과 평가 결과 (총 000개의 data set 중에서 train : 000개, test : 000개로 진행)

```
calculation mAP (mean average precision)...\nDetection layer: 139 - type = 28\nDetection layer: 150 - type = 28\nDetection layer: 161 - type = 28
```

```
76\n\nDetection count = 250, unique truth count = 120\nstreet_id = 1, name = street, ap = 39.59 (TP = 150, FP = 59)\nclass_id = 1, name = street, ap = 100.00 (TP = 150, FP = 0)\nclass_id = 2, name = street, ap = 75.59 (TP = 10, FP = 0)\nclass_id = 3, name = tree, ap = 73.59 (TP = 10, FP = 0)\nclass_id = 4, name = flowerpot, ap = 39.59 (TP = 10, FP = 0)\nclass_id = 5, name = plant, ap = 39.59 (TP = 10, FP = 0)\nclass_id = 6, name = street, ap = 39.59 (TP = 10, FP = 0)\nclass_id = 7, name = street, ap = 39.59 (TP = 10, FP = 0)\nclass_id = 8, name = antenna, ap = 0.00 (TP = 0, FP = 0)\nclass_id = 9, name = car, ap = 87.94 (TP = 23, FP = 0)\nclass_id = 10, name = car, ap = 77.97 (TP = 38, FP = 0)
```

```
for conf thresh = 0.25, threshold = 0.5, recall = 0.47, f1 score = 0.43,\nfor conf thresh = 0.25, TP = 100, FP = 0, mAP = 64.9%
```

```
for threshold = 0.50, used in each layer during train, each image recall\nmean average precision (mAP0.5) = 0.7032, or 70.31%\nTotal detection time: 3 seconds
```

```
calculation mAP (mean average precision)...\nDetection layer: 139 - type = 28\nDetection layer: 150 - type = 28\nDetection layer: 161 - type = 28
```

```
76\n\nDetection count = 250, unique truth count = 120\nstreet_id = 1, name = street, ap = 39.59 (TP = 150, FP = 59)\nclass_id = 1, name = street, ap = 100.00 (TP = 150, FP = 0)\nclass_id = 2, name = street, ap = 75.59 (TP = 10, FP = 0)\nclass_id = 3, name = tree, ap = 73.59 (TP = 10, FP = 0)\nclass_id = 4, name = flowerpot, ap = 39.59 (TP = 10, FP = 0)\nclass_id = 5, name = plant, ap = 39.59 (TP = 10, FP = 0)\nclass_id = 6, name = street, ap = 39.59 (TP = 10, FP = 0)\nclass_id = 7, name = street, ap = 39.59 (TP = 10, FP = 0)\nclass_id = 8, name = antenna, ap = 0.00 (TP = 0, FP = 0)\nclass_id = 9, name = car, ap = 87.94 (TP = 23, FP = 0)\nclass_id = 10, name = car, ap = 77.97 (TP = 38, FP = 0)
```

```
for conf thresh = 0.25, threshold = 0.5, recall = 0.47, f1 score = 0.43,\nfor conf thresh = 0.25, TP = 100, FP = 0, mAP = 64.9%
```

```
for threshold = 0.50, used in each layer during train, each image recall\nmean average precision (mAP0.5) = 0.7032, or 70.31%\nTotal detection time: 3 seconds
```

<정면 사진 testset 결과>

Object Detection의 성능평가에서 주로 사용되는 MS COCO Dataset에서 기존 YOLOv4의 성능은 AP = 43.0%, mAP0.5 = 64.9% 정도의 성능을 보이고 있음 (성능 평가 기준)

위의 최종 학습된 모델의 결과를 확인해 보면 여러 각도의 building image를 test set으로 한 결과는 window AP = 69.30%, mAP0.5 = 23.97%의 성능을 보이고 있으며, 과업 범위에 해당하는 정면에서 촬영한 building image의 test set에 대해서는 **window AP = 98.59%, mAP0.5 = 70.31%**의 좋은 성능을 확인 할 수 있음

선행 연구인 Ali, Haider, et al. "Window detection in facades," 14th International Conference on Image Analysis and Processing (ICIAP 2007). IEEE, 2007의 facade 건물의 window detecting 결과를 비교해 보면 **coverage 50의**

precision을 환산하면 **87.96%** 이고, 본 모델의 window에 대한 **precision**은 **98.59%** 아므로 월등한 성능을 보이는 것을 확인 할 수 있음

<여러 각도의 testset 결과>

04 한계점 및 발전 방향

4. 한계점 및 발전 방향

4-1. 한계점

- 정면 위주의 데이터

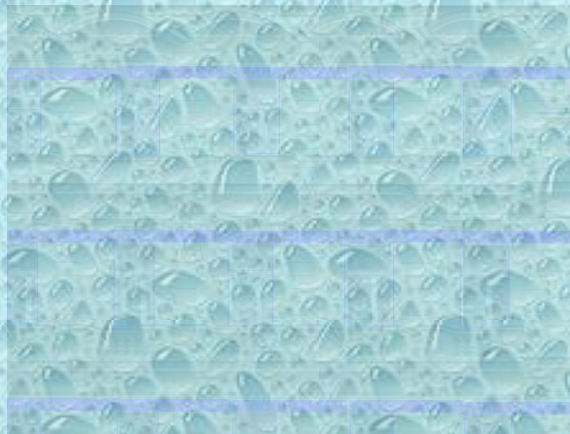
- 측면에서의 detection 정확도가 다소 감소되는 경향을 극복하는 방안 마련

통유리 방식의 한국형 데이터

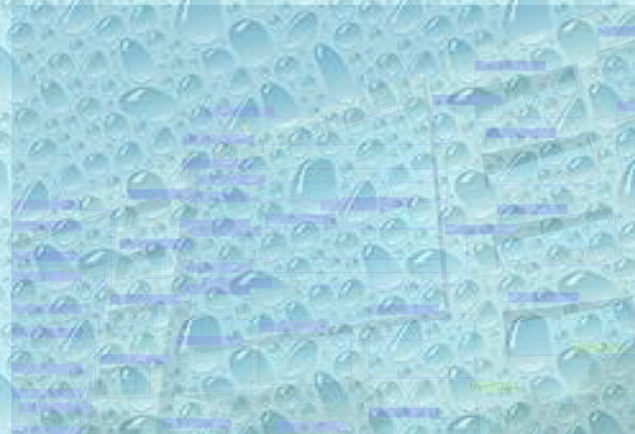
→ AI 모델을 학습시킬 충분한 한국형 데이터 확보 필요

→ 연속되는 유리 형태에서 창문에 대한 정확한 재정의 필요

건축물 다양성 대응을 위한 세그멘테이션 기반 인식 모델 활용 필요



<Facade image>



<한국형 image>



<segmentation image>

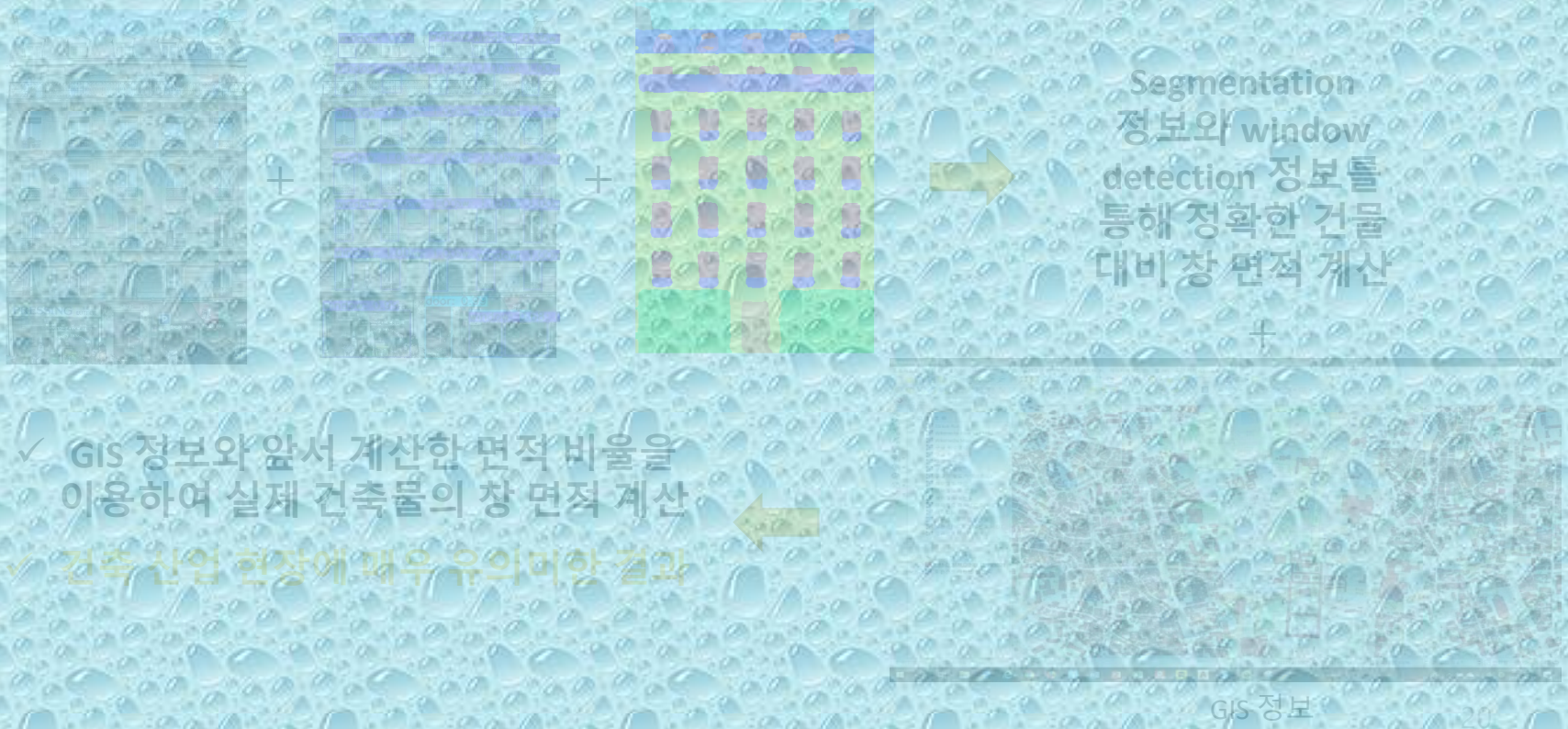
4. 한계점 및 발전 방향

4-2. 향후 연구 발전 방향

- 건물 대비 창문의 비율 계산의 정확도 향상

segmentation 결과와 detecting된 window의 면적을 이용하여 정확한 입력 이미지
의 건물 대비 창문 비율 계산을 위해 DeepLab v3 model 고도화 진행

- 건축물의 GIS 정보를 대입하여 실제 건물의 면적을 유추하고 실제 건물의 창문 면적
계산 진행



감사합니다