

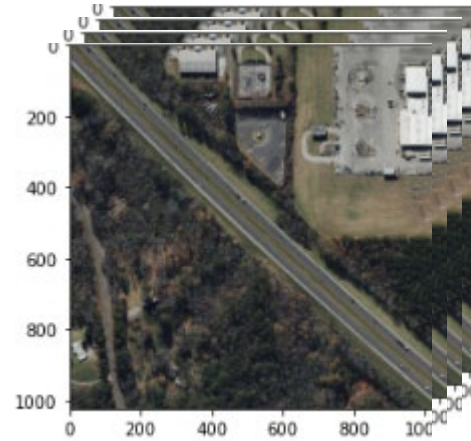
# DACON Building Segmentation

---

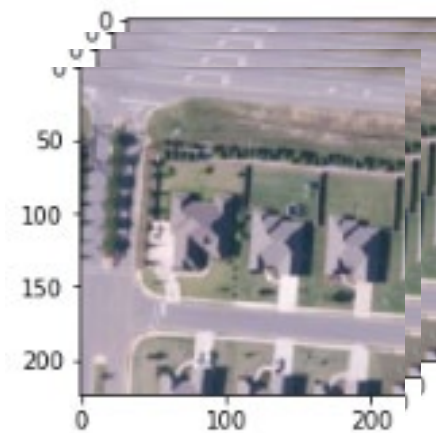
Team – AIAC Lab

Younghoon Na  
Email : [nayounghoon0223@gmail.com](mailto:nayounghoon0223@gmail.com)  
Github : [younghoonNa@github.com](https://github.com/younghoonNa)

# Data Processing



Train Dataset  
 $3 \times 1024 \times 1024$



Test Dataset  
 $3 \times 224 \times 224$

# Data Processing



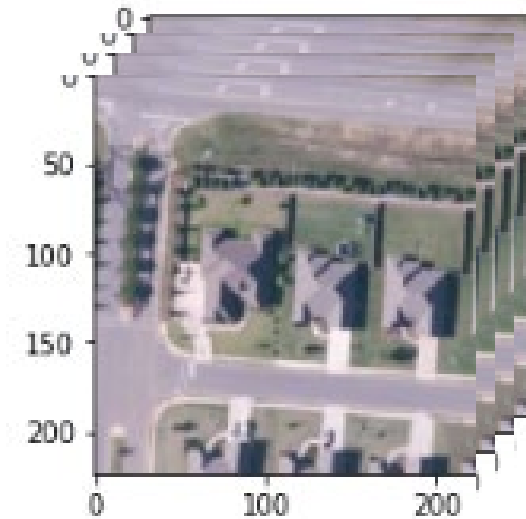
Train Dataset [256]

$16 \times 3 \times 256 \times 256$



Train Dataset [192]

$36 \times 3 \times 192 \times 192$



Test Dataset

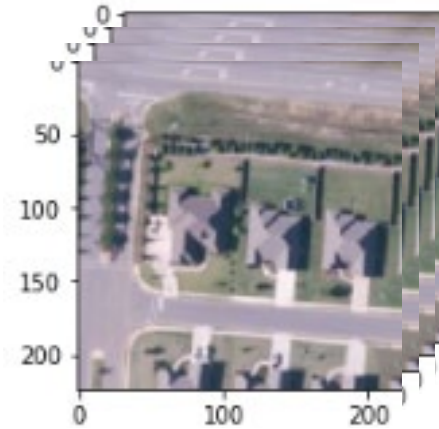
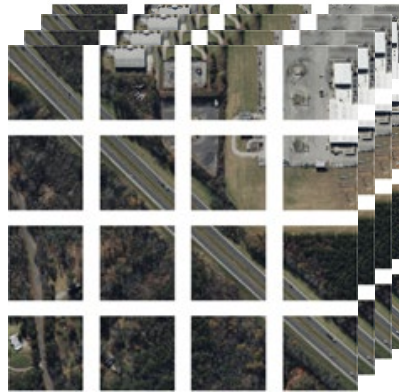
$3 \times 224 \times 224$

# Experimental Environment

Train Dataset [256]

$3 \times 256 \times 256$

Total: 114240장



Test Dataset

$3 \times 224 \times 224$

Total: 60640장

Train 44%

Val 22%

Test 34%

Experiment

Train 80%

Val 20%

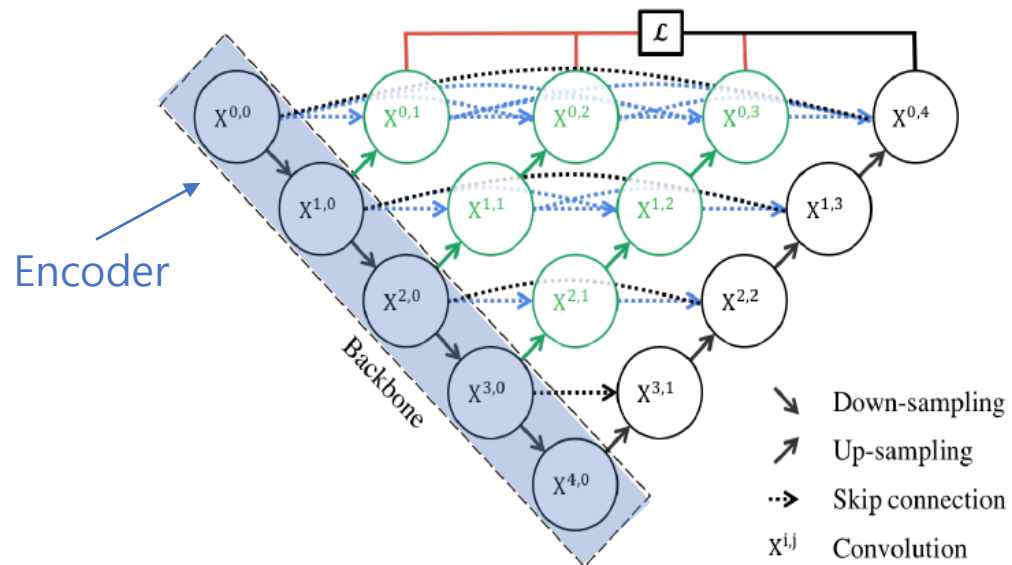
Training

Train 100%

Final Training

# Model & Augmentation

Unet++ Architecture



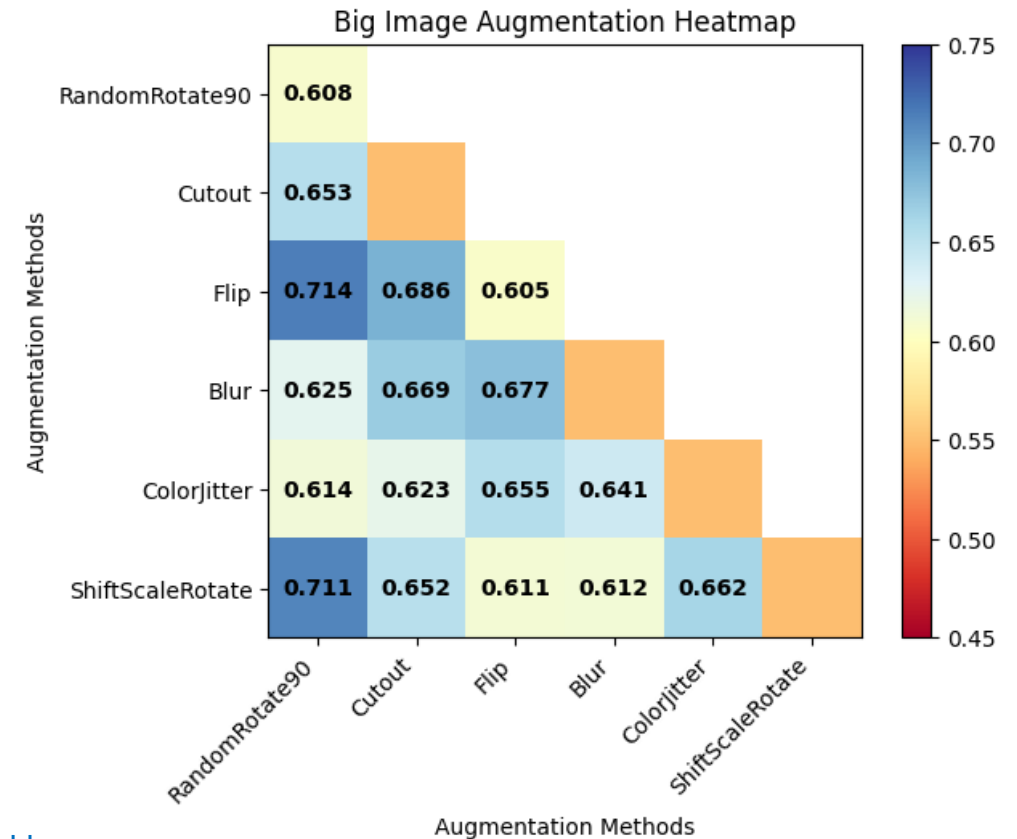
- Segmentation Pytorch 라이브러리를 바탕으로 하여, Unet++ Decoder를 기본으로 하는 모델을 사용

Encoder	Decoder	Test Dice
se_resnext50_32x4d	FPN	71.400
	DeepLabV3Plus	69.049
	DeepLabV3	66.985
	PAN	70.029
	Unet	72.285
	Unet++	73.169

Encoder	Decoder	Test Dice
resnet18	Unet++	71.912
se_resnext50_32x4d		73.169
se_resnext101_32x4d		75.029
timm-efficientnet-b2		71.577
timm-efficientnet-b3		72.378
timm-efficientnet-b4		73.132

# Loss & Augmentation

Loss1	Loss2	Rate	Test Accuracy
Dice	BCE	8 : 2	68.729
	BCE	7 : 3	68.598
	BCE	6 : 4	67.684
	Focal	8 : 2	72.087
	Focal	7 : 3	71.685
	Focal	6 : 4	71.866
	Tversky	8 : 2	69.637
	Tversky	7 : 3	68.605
	Tversky	6 : 4	67.264

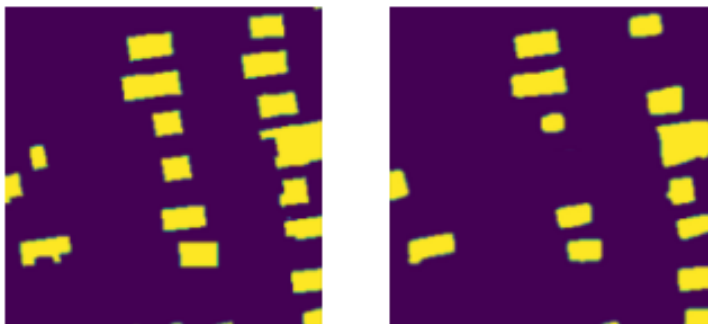


- Test Score Matrix가 Dice Score이므로 Dice를 기반으로 하는 Loss Function 구현
- Train Dataset 이미지는 256x256, Test Dataset 이미지는 224x224 이므로 RandomCrop를 사용
- 3개 이상 Augmentation을 적용하였을 때 성능보다, 1~2개로 간단하게 Augmentation을 적용하였을 때 성능 향상

# Problems

1. 건물의 크기가 작은 경우, 모델이 인식을 못하는 상황 발생
2. 모델의 출력 값을 확인했을 때, 건물 내부가 비어 있는 출력 값이 발생
3. 건물의 경계선이 모호한 출력 발생
4. 16개의 이미지 분할로 인한 배경 정보 손실 문제

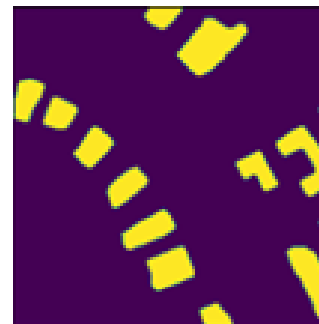
문제 1



문제 2



문제 3



문제 4

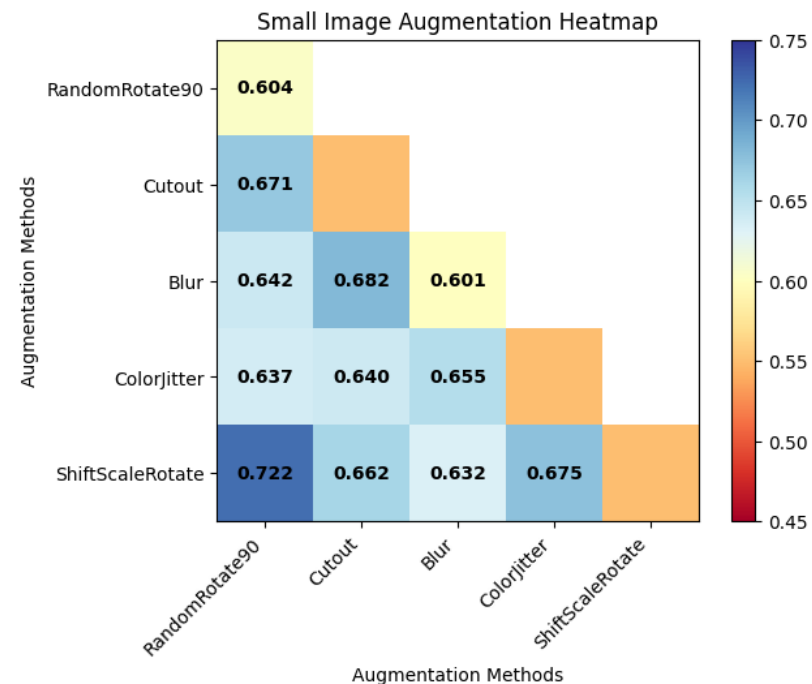
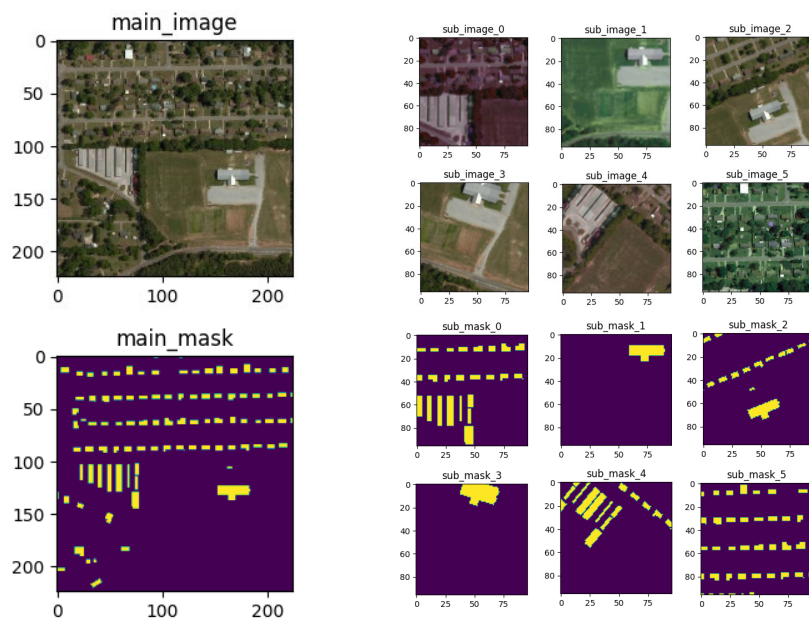


# Method1

## 건물의 크기가 작은 경우, 모델이 인식을 못하는 문제

- Self-Supervised Learning 방법론 중 하나인 DINO에서 96x96 크기의 이미지를 잘라서 모델의 넣는 것에 영감을 받아서 진행
- 96x96 크기의 이미지는 원본 이미지의 Augmentation보다 더 강한 Augmentation 적용

### 원본 이미지와 Crop된 이미지

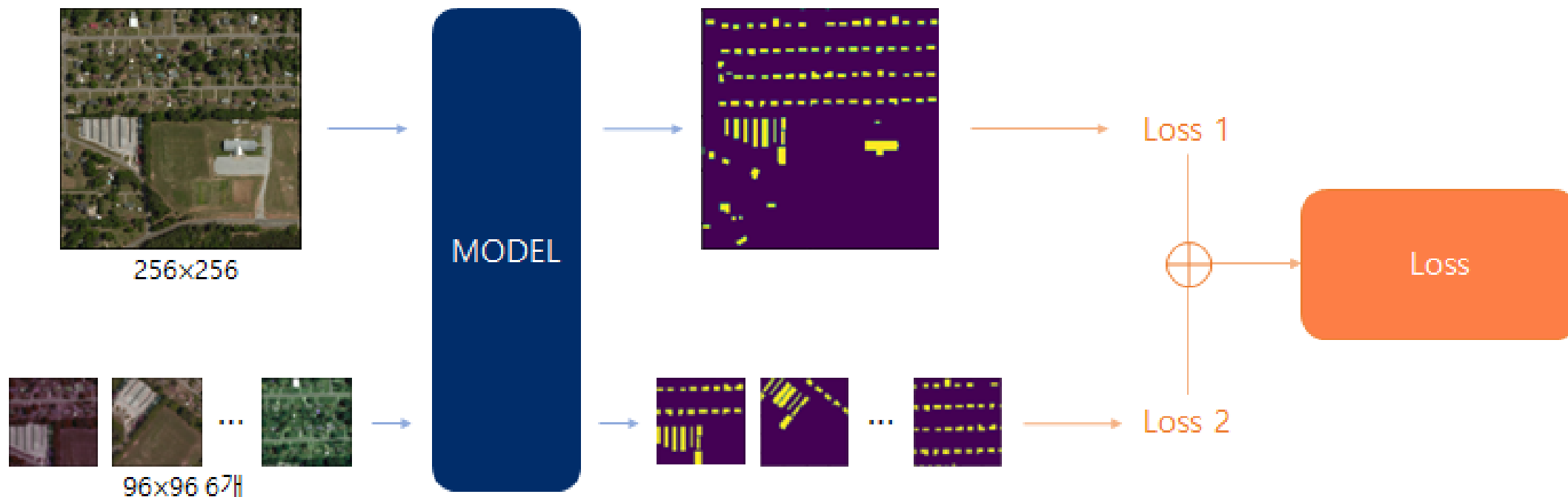




# Method1

## 건물의 크기가 작은 경우, 모델이 인식을 못하는 문제

- 256x256 이미지의 Loss와 96x96 이미지 6개에 대한 Loss의 비율은 5:5로 설정

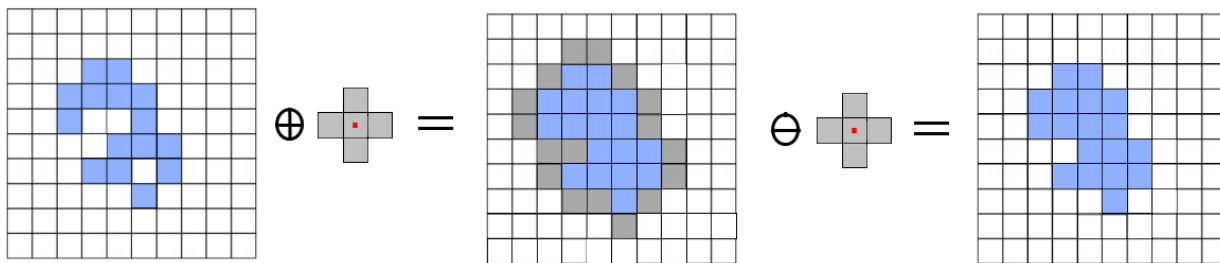


# Method2

## 모델의 출력 값을 확인했을 때, 건물 내부가 비어 있는 출력 값이 발생

- 모폴로지 침식, 팽창 기술을 활용하여 모델의 출력물의 구멍을 제거

모폴로지 닫힘 연산



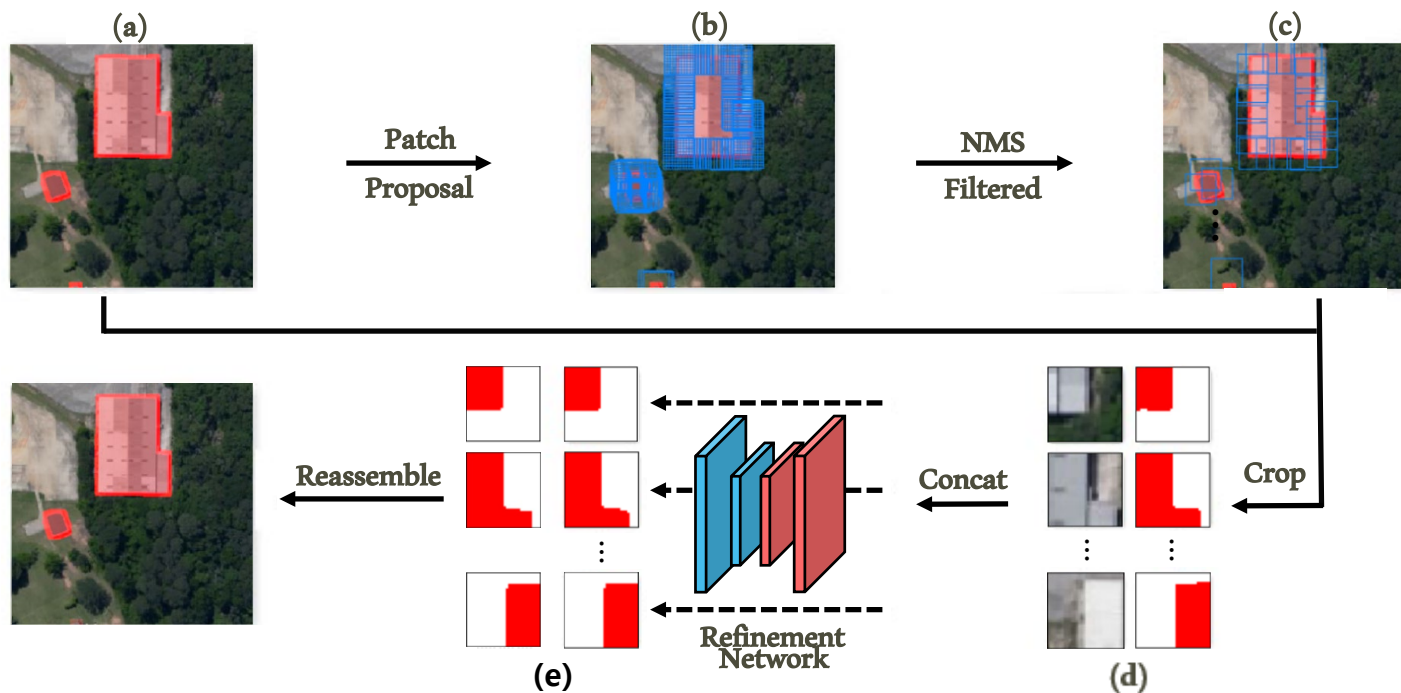
적용 전과 적용 후



# Method3

## 건물의 경계선이 모호한 출력 발생

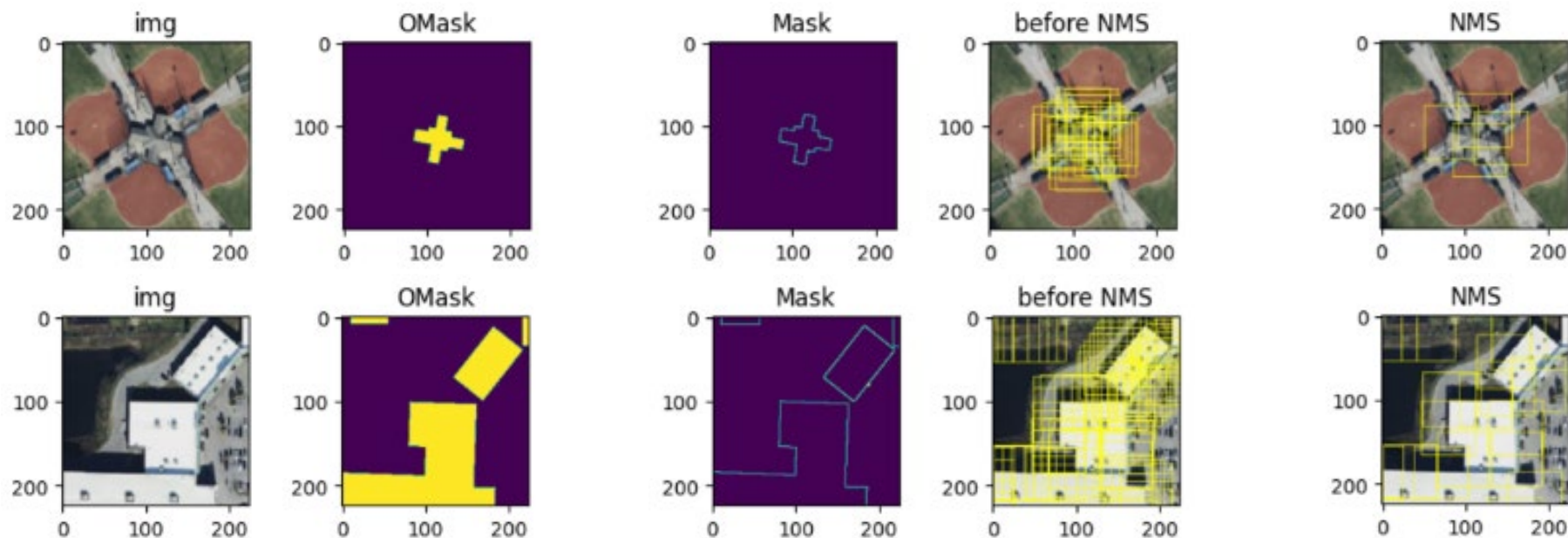
- Cityscapes Dataset에서 1위를 달성한 방법론 Boundary Patch Neural Network의 방법론을 적용



- (a) 모델의 출력
- (b) 경계선을 기준으로 패치 생성
- (c) Non Maximum Suppression 알고리즘 적용
- (d) 모델의 Output과 실제 이미지의 경계선 비교
- (e) Refinement Network 통과한 Output과 정답 비교

# Method3

## 건물의 경계선이 모호한 출력 발생



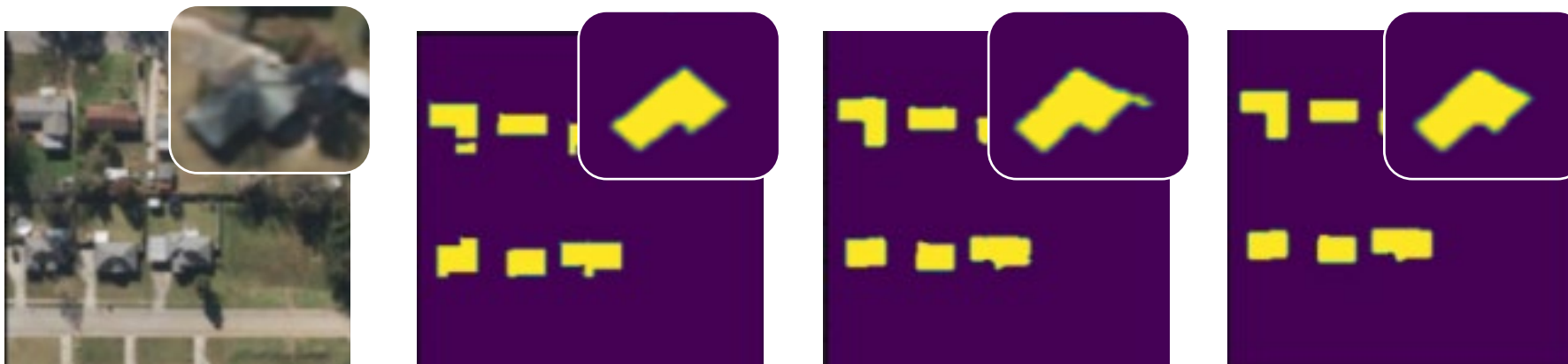
모폴로지 팽창 연산 적용 후,  
원본 마스크를 빼는 작업을 통해 경계선 출력

`torch.where()` 함수를 통해  
경계선을 기준으로 패치 함수 적용

0.3의 Threshold 값을 적용하여  
Non-Maximum Suppression 알고리즘 적용

# Method3

건물의 경계선이 모호한 출력 발생



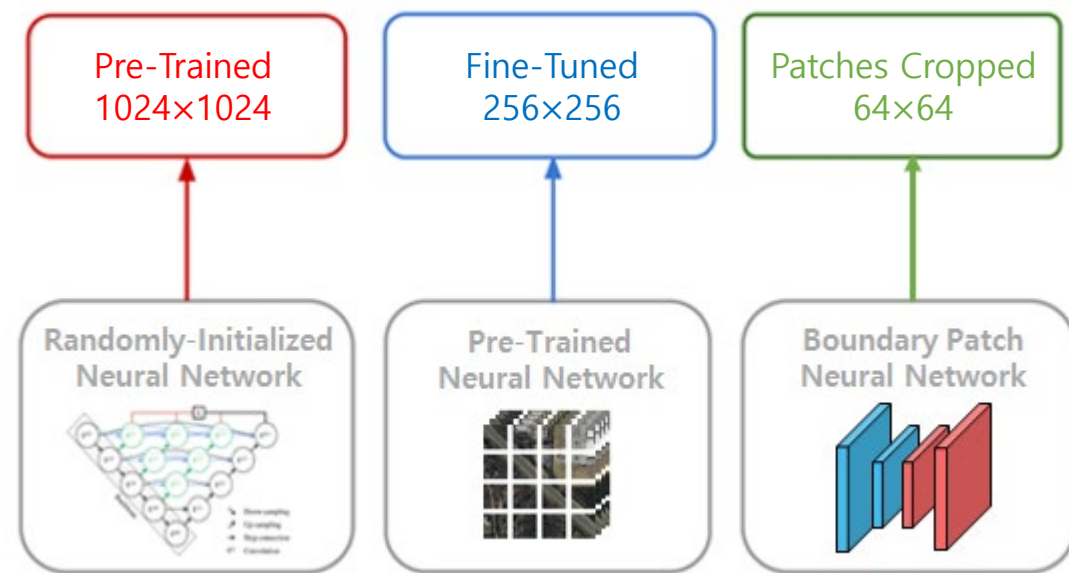
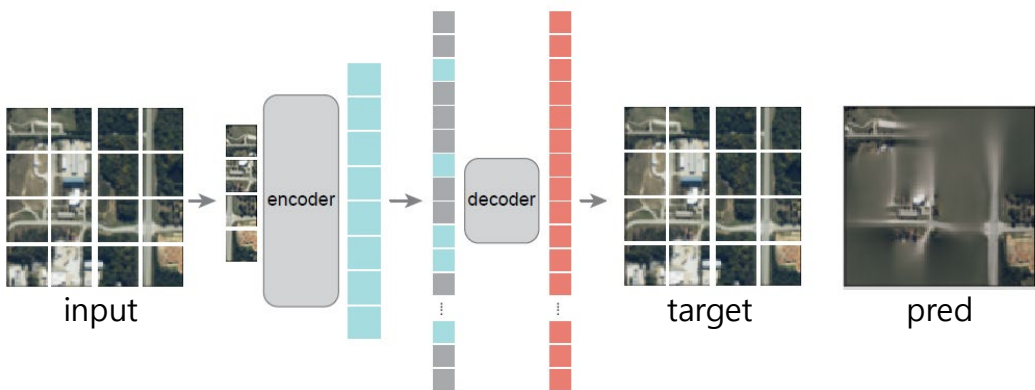
단일 모델 기준, Public Score 0.004 상승

0.8181890831 → 0.8185862116

# Method4

## 16개의 이미지 분할로 인한 배경 정보 손실 문제

1. 최근 각광받는 Masked Image Modeling 방식을 통해 고해상도 이미지 정보를 기억하는 Pre-Training Model을 만들어 저해상도 이미지를 이용하여 Fine-Tuning하여, 보다 건물 정보를 잘 식별하는 모델을 만들기를 희망
2. Masked Image Modeling 방식은 Transformer 모델에서만 동작, 하지만 Transformer 모델들의 성능은 이번 과제에서 대체적으로 낮음
3. 따라서 1024x1024 이미지에서 Unet++ 모델로 Initialized된 모델을 사용하여 Fine-tuning 진행



## 7개의 모델을 바탕으로 Hard Voting 진행.

1. Segmentation Pytorch Library의 공식 문서를 보며 Segmentation Kaggle Competition에서 상위권 점수를 기록한 팀들의 앙상블 방법을 벤치마킹
2. 다양한 Decoder를 사용하여 다양한 관점에서 모델을 보고싶었지만, Unet++의외에는 좋은 성능을 보이는 Decoder 존재하지 않음
3. 따라서 Encoder 및 해상도를 다양하게 하려고 노력

모델 Encoder	모델 Decoder	해상도	Epoch	Pre-Trained Model Initialized
se_resnext_101_32x4d	Unet++	224x224	50	O
se_resnext_101_32x4d	Unet++	224x224	40	O
se_resnext_101_32x4d	Unet++	192x192	50	O
se_resnext_50_32x4d	Unet++	224x224	50	X
timm-efficientnet-b2	Unet++	224x224	50	O
timm-efficientnet-b3	Unet++	224x224	50	X
timm-efficientnet-b4	Unet++	224x224	50	X


# Experiments

## 최종 선택한 2개의 모델

1. 7개의 모델을 Hard Voting 진행 – 82.3319 (Public)
2. 7개의 모델에 대해 경계선 네트워크 적용 후 Hard Voting 진행 – 82.1237 (Public)

877382	Ensemble_7_Calibration_%7Cse_resnext101_50%7C_%7C40%7C_%7Cse_resnext101_192x192%7C_%7Cse_resnext50%7C_effi_b4_b3_b2.csv	2023-07-28 08:08:06	0.8212374861	✓
877344	Ensemble_7_%7Cse_resnext101_50%7C_%7C40%7C_%7Cse_resnext101_192x192%7C_%7Cse_resnext50%7C_effi_b4_b3_b2.csv	2023-07-28 06:01:15	0.8233190349	✓

## 최종 Private Score

8	Hallym_AIAC		0.82121	53	13일 전
---	-------------	---	---------	----	-------



*Thank you*