

# **GIANA**

**Gastrointestinal Image ANALys Challenges**

**이효준 심재형 이하영 김설아**

# OUR TEAM



이효준

팀장  
DUCK-Net



심재형

ResUNet++



이하영

TGA-Net



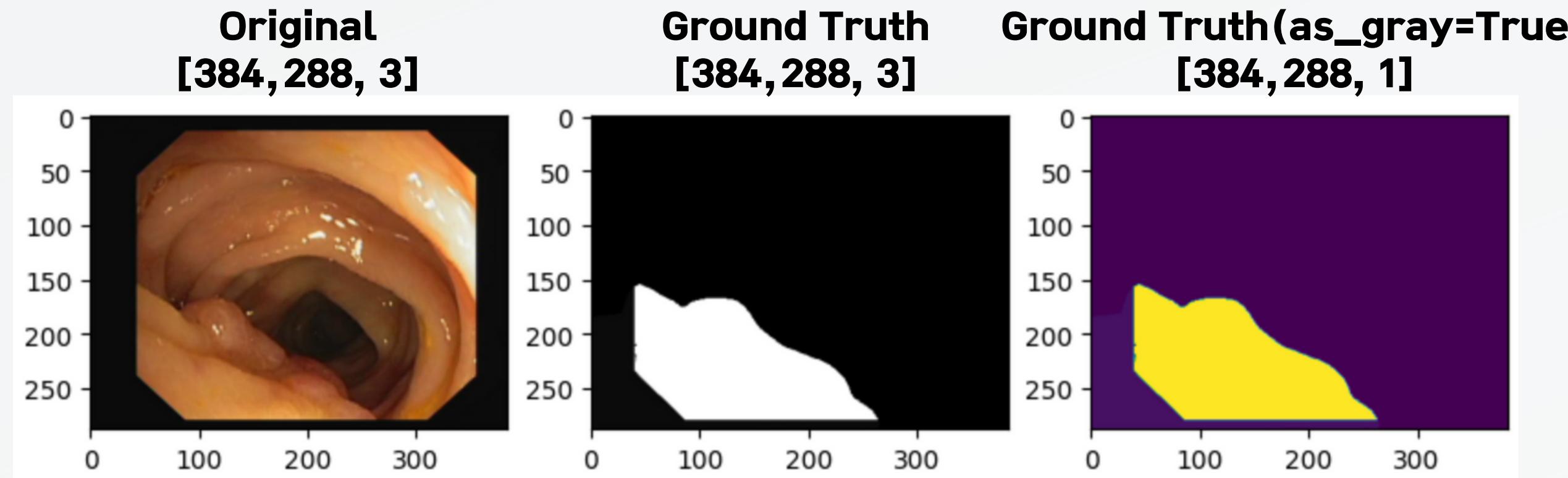
김설아

PEFNet

# CONTENT

- 01** 데이터 EDA 및 전처리
- 02** 모델 비교
- 03** 모델 선정
- 04** 최종 결과
- 05** 고찰
- 06** Q&A

# 데이터 EDA 및 전처리



class_names	r	g	b
background	0	0	0
polyp	255	255	255

**Polyp\_segmentation**

```
└ train
    └ PNG
        └ Originals(612)
        └ Ground Truth(612)
    └ TIF
        └ Originals
        └ Ground Truth
    └ test
        └ Originals(300)
        └ Ground Truth(300)
└ class_dict.csv
└ metadata.csv
└ README.md
```

# 데이터 EDA 및 전처리

**Data augmentation.** We implemented data augmentation on the training set, significantly improving the model’s generalization capabilities to the point where regularization techniques such as dropout were unnecessary. The library used to implement the augmentations is Albumentations<sup>39</sup>. This involved randomly applying transformations to the training images, resulting in significantly different variations from the original images and helping the model better generalize to unseen data.

Before each epoch, we randomly augmented the training input using augmentations inspired by previous work<sup>26</sup> but modified to fit the specific needs of our model. The augmentation techniques we used are:

1. Horizontal and vertical flips
2. Color jitter with a brightness factor uniformly sampled from [0.6, 1.6], a contrast of 0.2, a saturation factor of 0.1, and a hue factor of 0.01
3. Affine transforms with rotations of an angle sampled uniformly from  $[-180^\circ, 180^\circ]$ , horizontal and vertical translations each of a magnitude sampled uniformly from  $[-0.125, 0.125]$ , scaling of a magnitude sampled uniformly from [0.5, 1.5] and shearing of an angle sampled uniformly from  $[-22.5^\circ, 22^\circ]$ .

Out of these augmentations, the color jitter was applied only to the image, while the rest were applied consistently to both the image and the corresponding segmentation map.

# 데이터 EDA 및 전처리

```
def build_augmentation(is_train=True):
    if is_train: # 훈련 데이터
        return Compose([
            HorizontalFlip(p=0.5), # 50% 확률로 좌우 대칭
            VerticalFlip(p=0.5), # 50% 확률로 상하 대칭

            ColorJitter(
                brightness=(0.6, 1.6), # 밝기 요소를 0.6~1.6 범위에서 균등하게
                contrast=0.2, # 대비 0.2
                saturation=0.1, # 채도 0.1
                hue=0.01, # 색조 0.01
                always_apply=True
            ), # 색상 변환

            Affine(
                rotate=(-180, 180), # 회전 범위
                translate_percent=(-0.125, 0.125), # 이동 범위
                scale=(0.5, 1.5), # 확대/축소 범위
                shear=(-22.5, 22.5), # 기울임이기 범위
                always_apply=True), # 변환을 적용할 확률
            Resize(
                width=224,
                height=224
            )
        ])
    return Compose([ # 테스트 데이터의 경우, 크기 조절만 수행
        Resize(
            width=224,
            height=224
        )
    ])
])
```

- 수평 및 수직 뒤집기

## COLOR JITTER

- 밝기 요소 (0.6, 1.6) 범위에서 균등하게 샘플링.
- 대비 0.2
- 채도 0.1
- 색조 0.01

## AFFINE TRANSFORMS

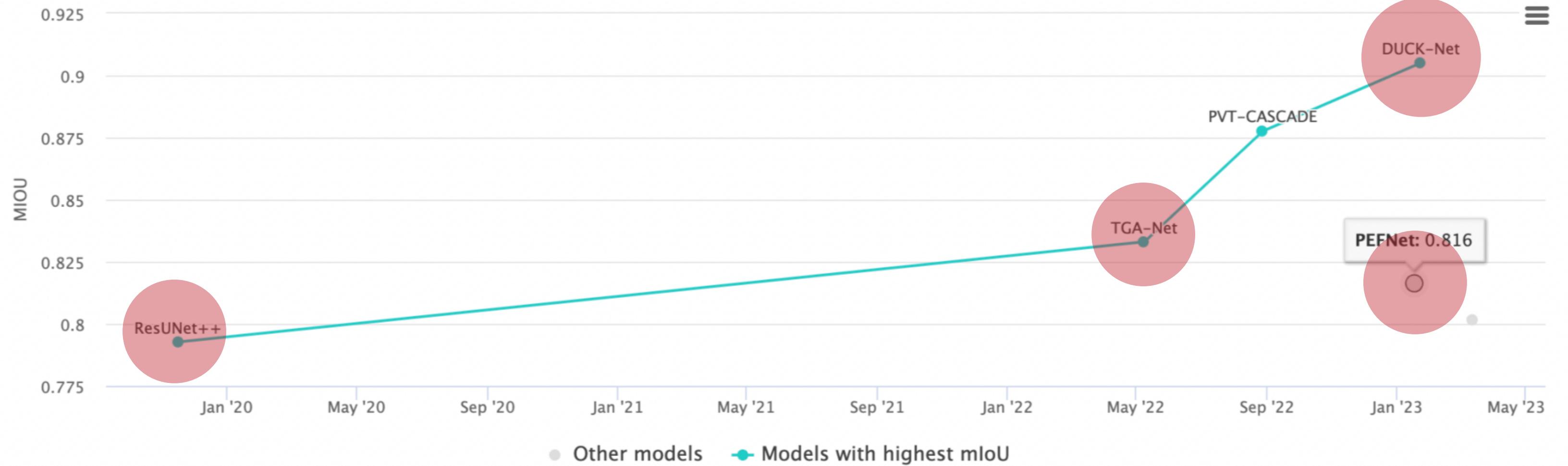
- 회전 각도  $[-180^\circ, 180^\circ]$  범위에서 균등하게 샘플링
- 수평 및 수직 이동  $[-0.125, 0.125]$  범위에서 균등하게 샘플링
- 크기 변환  $[0.5, 1.5]$  범위에서 균등하게 샘플링
- 기울임 변환  $[-22.5^\circ, 22^\circ]$  범위에서 균등하게 샘플링

# 모델 비교

## Polyp Segmentation on Kvasir-SEG

Leaderboard      Dataset

View  by



## 평가지표 - IoU

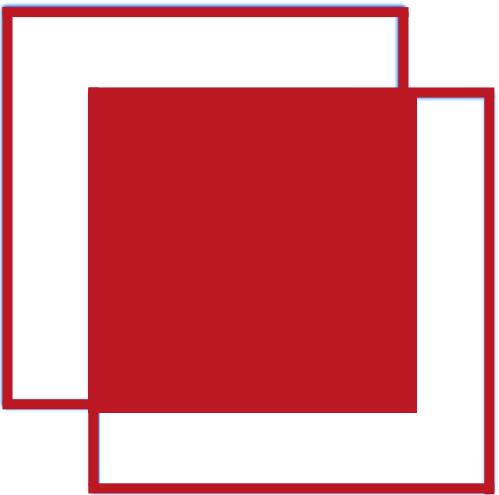
# 모델 비교

### Intersection over union

- 2개(정답, 예측)의 영역이 '얼마나 겹쳐져 있는가'
- 0~1
- => 0.7

Area of Intersection

$$IoU = \frac{\text{Intersection over Union}}{\text{Area of Union}}$$



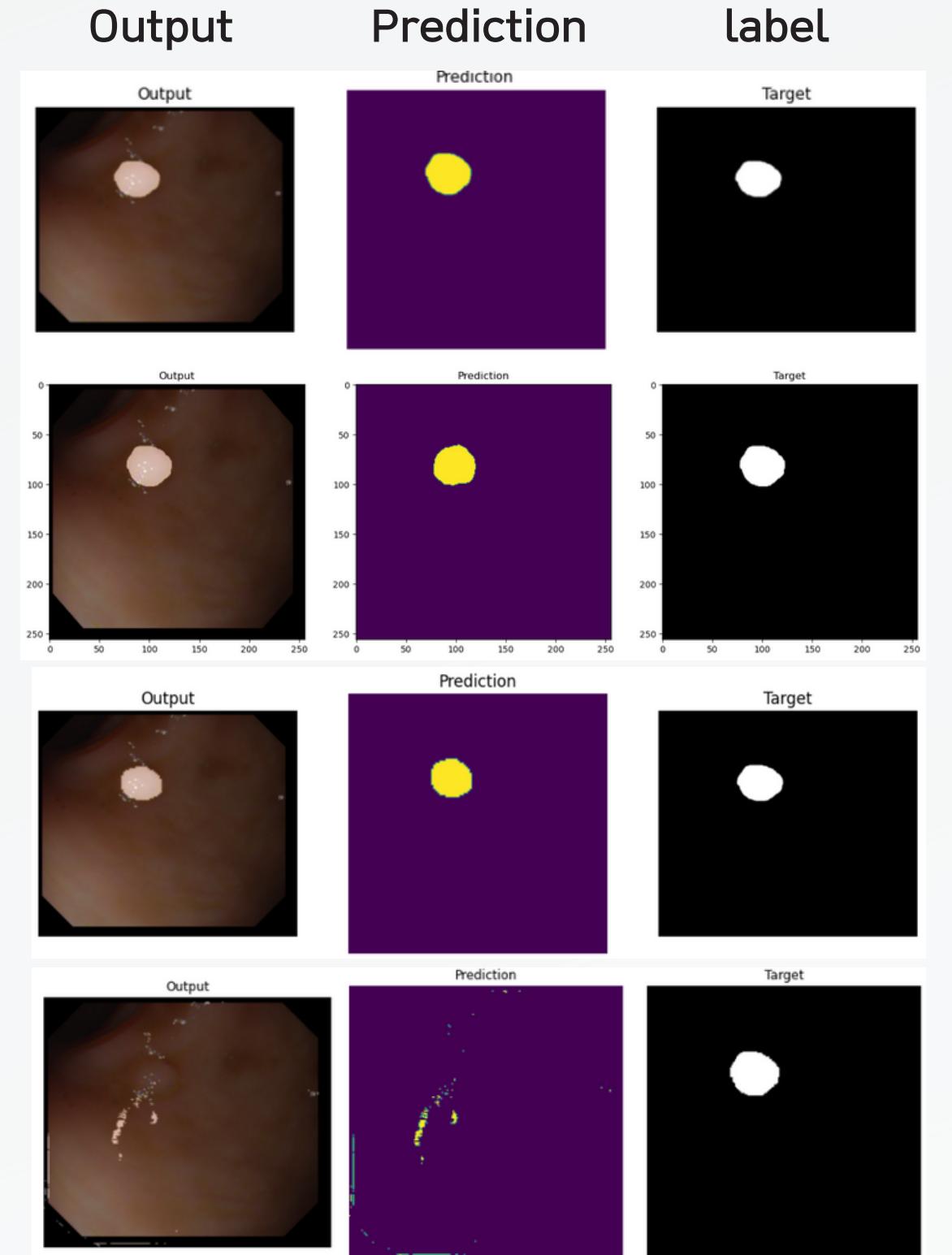
# 모델 비교

(Transfer Learning)  
**DUCK-Net**  
IoU : 0.9162

**TGA-Net**  
IoU : 0.9143

**PEFNet**  
IoU : 0.8427

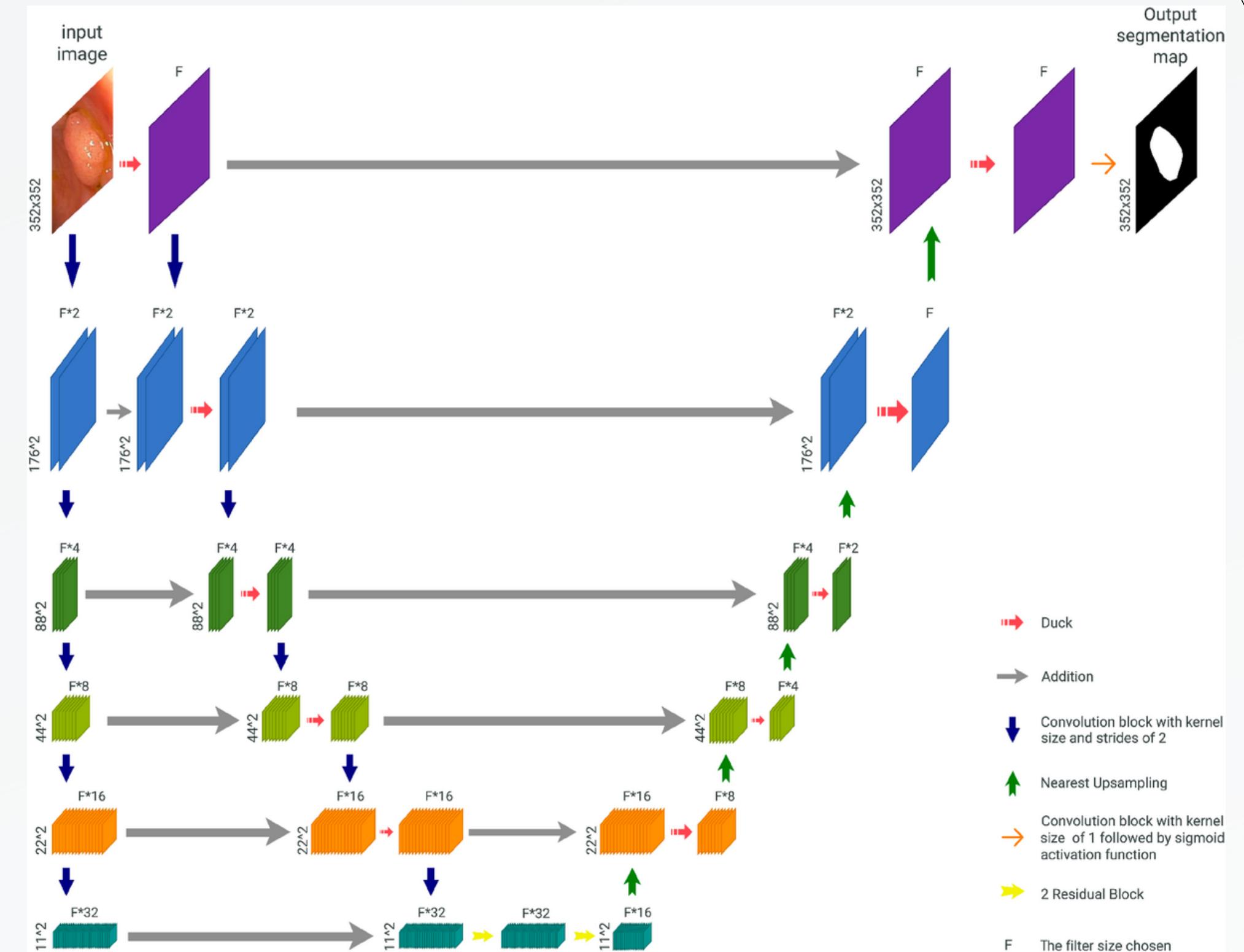
**ResUNet++**  
IoU : 0.005



# DUCK-Net

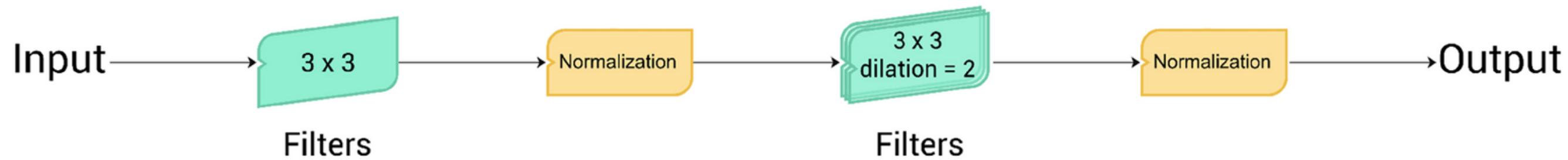
# 모델 선정

- U-Net의 기본 구조에  $3 \times 3$  conv. blocks에서 마지막 블록 제외, DUCK block으로 대체
- DUCK block으로 인한 세부 정보 손실로 인한 문제를 해결하기 위해 컨볼루션 처리 구현하지 않는 secondary down-sampling
- concatenation 대신 addition을 사용

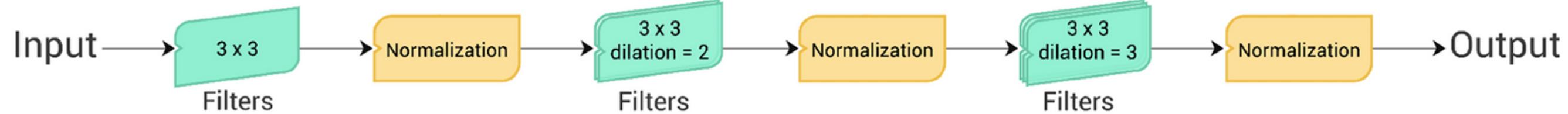


# 모델 선정

## Midscope Block



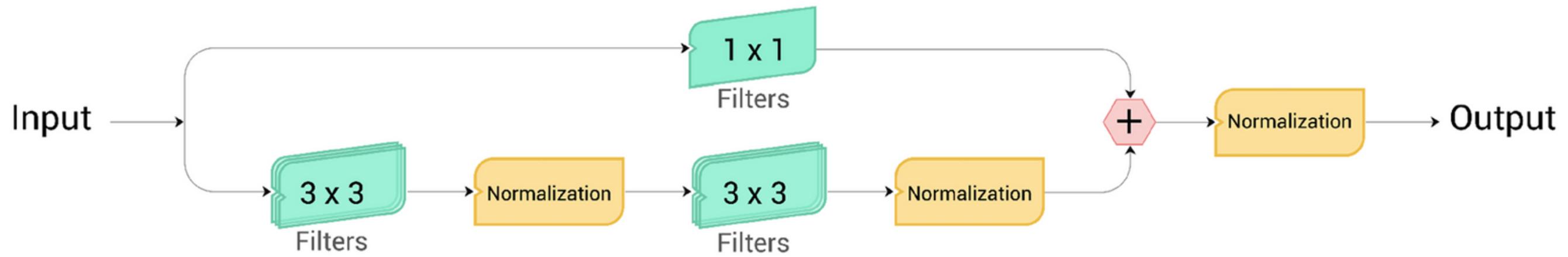
## Widescope Block



- 필요한 매개변수를 줄이고 네트워크가 higher-level feature를 더 잘 이해하도록 함
- 두드러진 특징을 학습하는 것을 목표로 함
- Midscope : 7x7, Widescope : 15x15 크기의 커널을 시뮬레이션

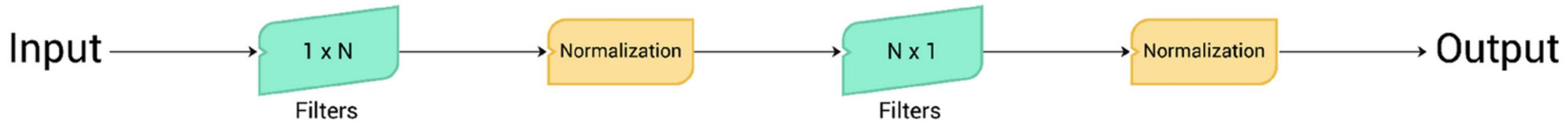
# 모델 선정

## Residual Block



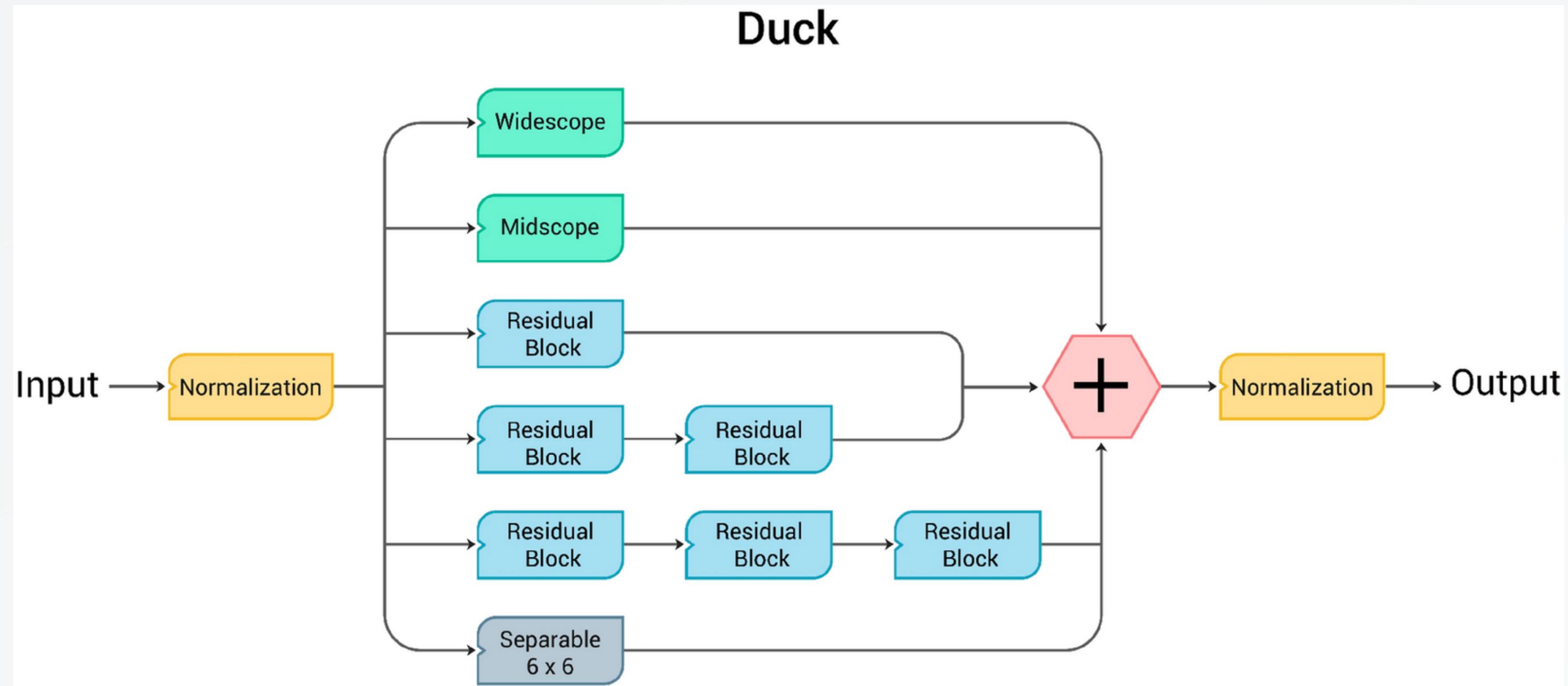
- 각각 1, 2, 3개의 residual blocks 조합을 사용해 각 커널의 크기(5, 9, 13)를 시뮬레이션

## Separated Block



- 1xN과 Nx1 커널을 결합하면 NxN 커널과 유사하게 동작한다는 아이디어에서 착안
- loss of diagonality는 다른 블록에서 보정하도록 설계

# 모델 선정



- 앞선 block을 결합하여 병렬로 사용되어 네트워크가 가장 중요한 부분을 정확하게 훈련할 수 있게 함
- 커널의 크기가 다양하다는 것은 general area를 찾는 동시에 edge도 정확하게 찾을 수 있다는 것
- lower-level과 high-level의 feature를 동시에 찾을 수 있음

# 고찰

	<b>DUCK-Net (paper / our)</b>	<b>TGA-Net (paper / our)</b>	<b>PEFNet (paper / our)</b>	<b>ResUNet++ (paper / our)</b>
<b>학습데이터</b>	<p>The Kvasir-SEG dataset (size 46.2 MB) contains 1000 polyp images and their corresponding ground truth from the Kvasir Dataset v2. The resolution of the images contained in Kvasir-SEG varies from 332x487 to 1920x1072 pixels.</p>			
<b>batch_size</b>	4(2)	16(8)	128(16)	8(16)
<b>epochs</b>	600(10)	500(25)	500(500)	120(200)

# 고찰

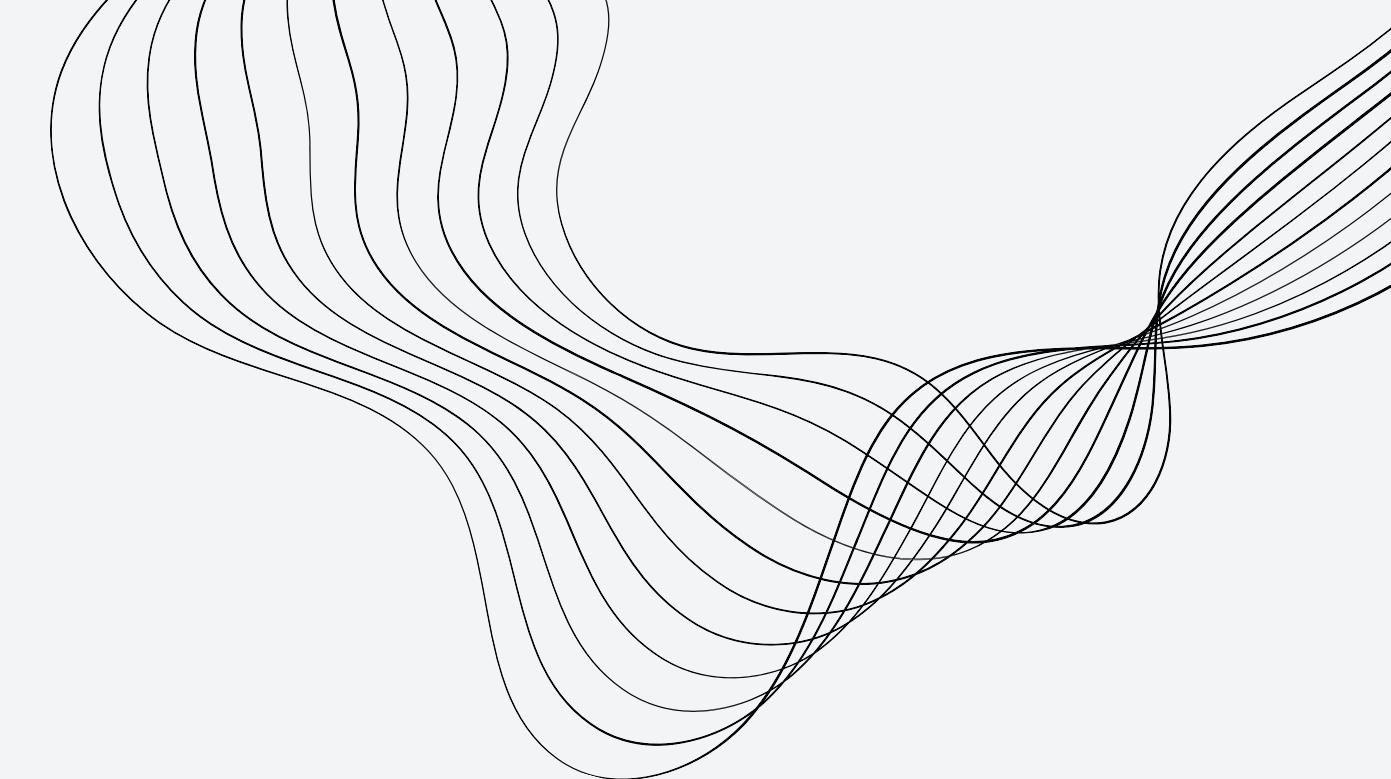
학습 데이터 셋 또는 학습 환경에 알맞는 모델을 사용하는 것이 중요

상식 수준의 최소 결과를 넘어야 함

pre-trained 모델을 사용하면  
더 좋은 결과를 낼 수 있었을 것으로 예상

배경과 경계가 불분명한 용종 감지 시 성능 제한이 확인되므로  
이에 대한 개선 방안이 필요

# 고찰



## DuckNet



꽥꽥 복날에 저를 먹지 말아주세요.

## PEFNet



가중치 파일 올려주세요 😞...

깃허브에 올라와 있는 코드를 조금 수정해야했어요

## ResUNet++



널 만나기전까지 내가 잘하는줄 알았어

## TGA-Net



torch는 쓰지 말자.



## 참고문헌

- [1] Zoph, Barret, et al. "Rethinking pre-training and self-training." *Advances in neural information processing systems* 33 (2020): 3833–3845.
- [2] <https://paperswithcode.com/sota/polyp-segmentation-on-kvasir-seg>
- [3] Dumitru, Razvan-Gabriel, Darius Peteleaza, and Catalin Craciun. "Using DUCK-Net for polyp image segmentation." *Scientific Reports* 13.1 (2023): 9803.
- [4] Rahman, Md Mostafijur, and Radu Marculescu. "Medical image segmentation via cascaded attention decoding." *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023.
- [5] Tomar, Nikhil Kumar, et al. "TGANet: Text-guided attention for improved polyp segmentation." *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Cham: Springer Nature Switzerland, 2022.
- [6] Nguyen-Mau, Trong-Hieu, et al. "Multi Kernel Positional Embedding ConvNeXt for Polyp Segmentation." *2022 RIVF International Conference on Computing and Communication Technologies (RIVF)*. IEEE, 2022.

**THANK'S  
FOR  
WATCHING**

