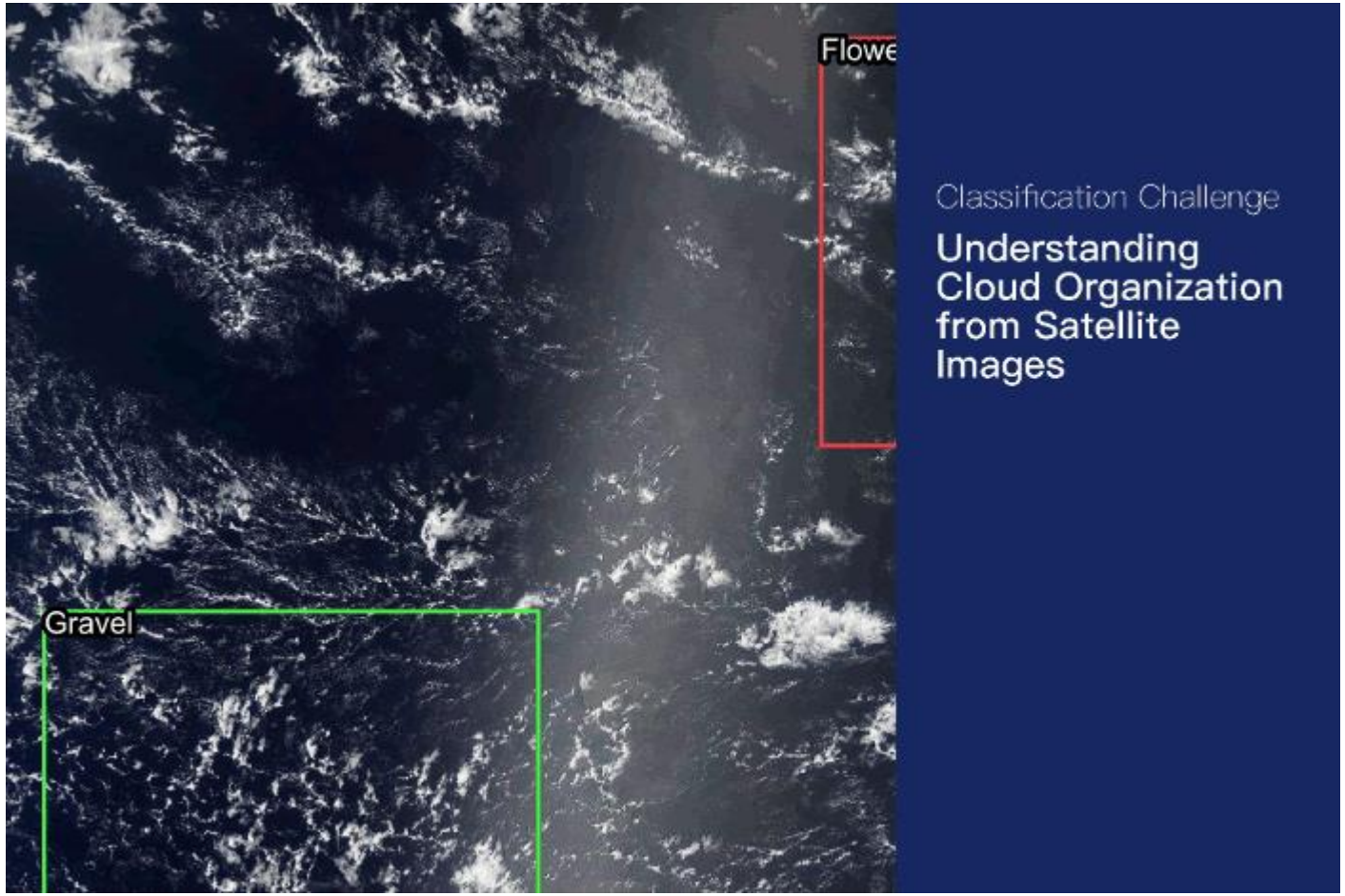
1. Overview

- (1) 대회 소개

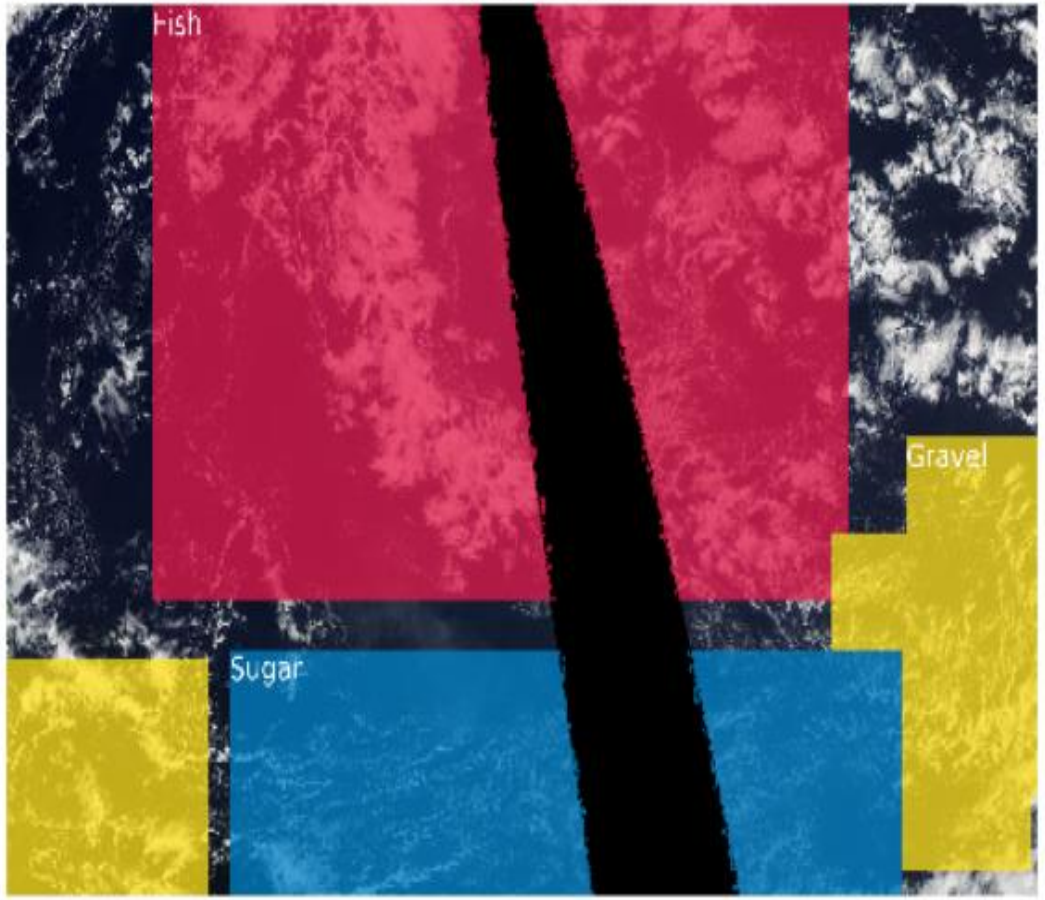


목적 : Help us remove the haze from climate models and bring clarity to cloud identification.

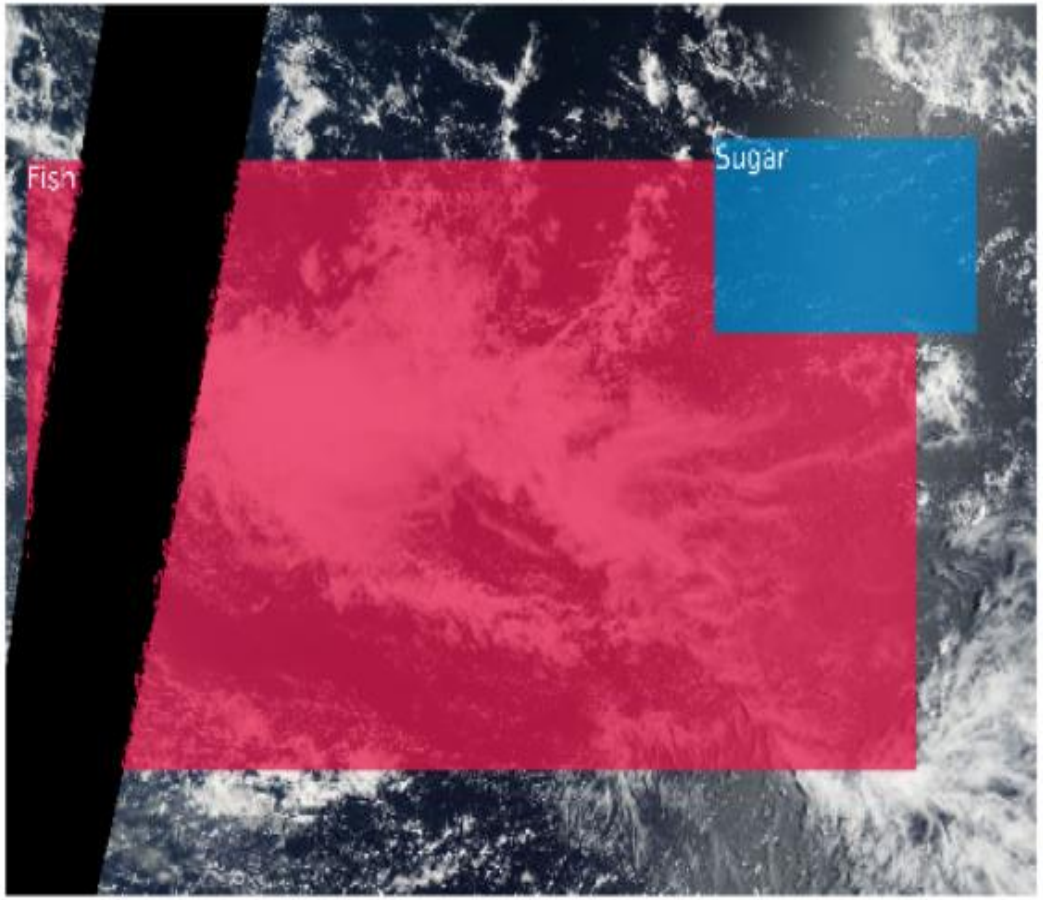
1. Overview

- (2) 입력 데이터

Fish Gravel Sugar



Fish Sugar



2. Solution

(1) 1st Solution Summary

1st soltuion

Network

- Model A: Unet with classification head
- Model B: FPN or Unet, no classification head

Backbones

- Resnet34
- Efficientnet-b1
- Resnext101_32x8d_wsl
- Resnext101_32x16d_wsl

2. Solution

(1) 1st Solution Summary

1st soltuion

Loss

- Classification part: BCE
- Segmentation part: BCE * 0.75 + DICE * 0.25

Optimizer

- AdamW, weight decay 0.01
- encoder learning rate 0.000025
- decoder learning rate 0.00025
- OneCycle scheduler, shallow models 30 epochs, models 15 epochs

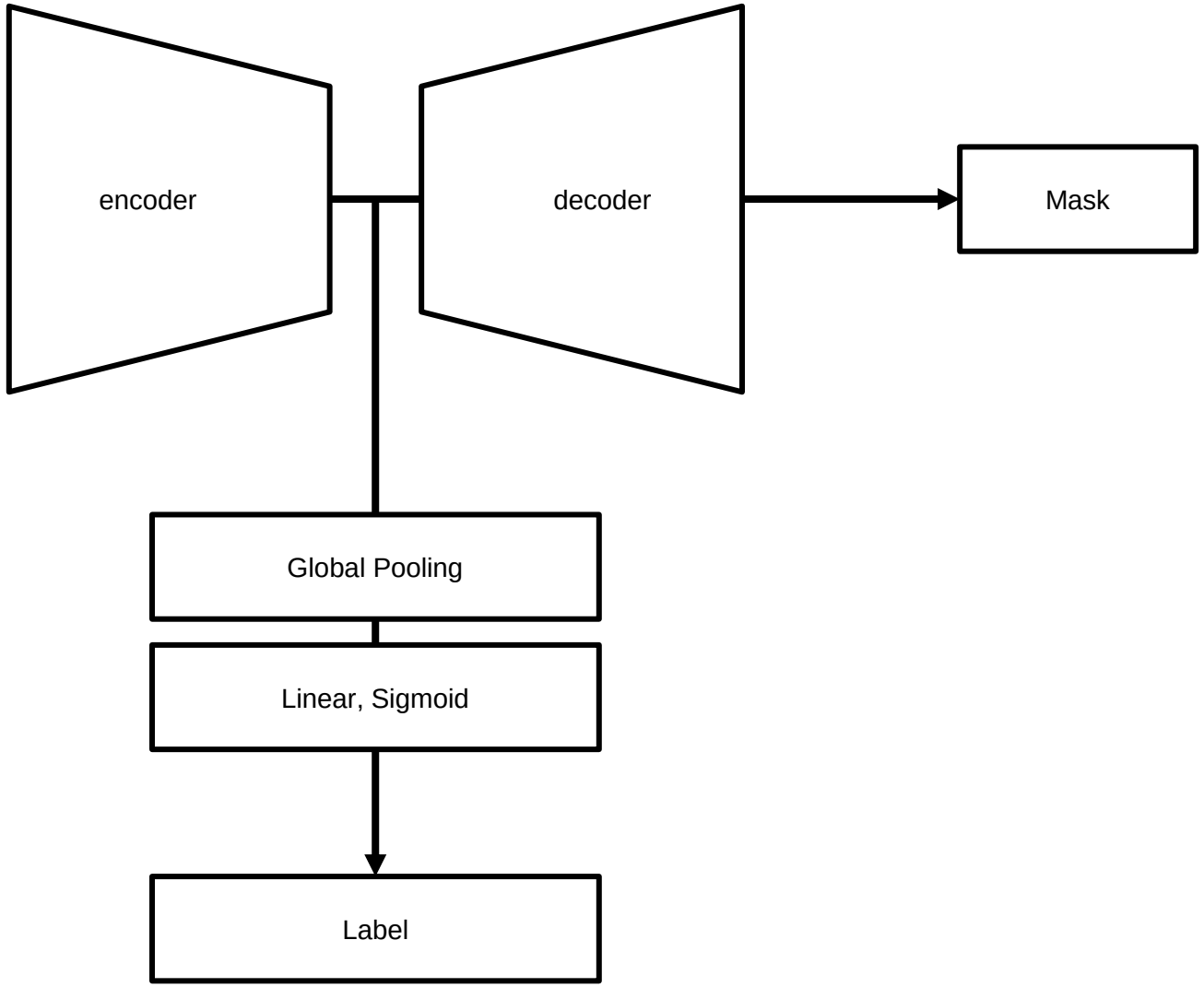
Augmentation

- Common: hflip, vflip, shift/scale/rotate, gid distortion, channel shuffle, invert, to gray
- Modal A: random crop, size 384
- Model B: full-size, size 384, 544, 576, 768

2. Solution

(1) 1st Solution Summary

1st soltuion



출처: pudae, “https://www.kaggle.com/c/understanding_cloud_organization/discussion/118080”

2. Solution

(1) 1st Solution Summary

1st soltuion

Other

- 추가적으로 Classification head와는 별도로 probability를 활용하여 segmentation을 classifier로 사용하였다.
- K개의 top probability pixel을 classification probability로 사용하였다. (ex K=17500)

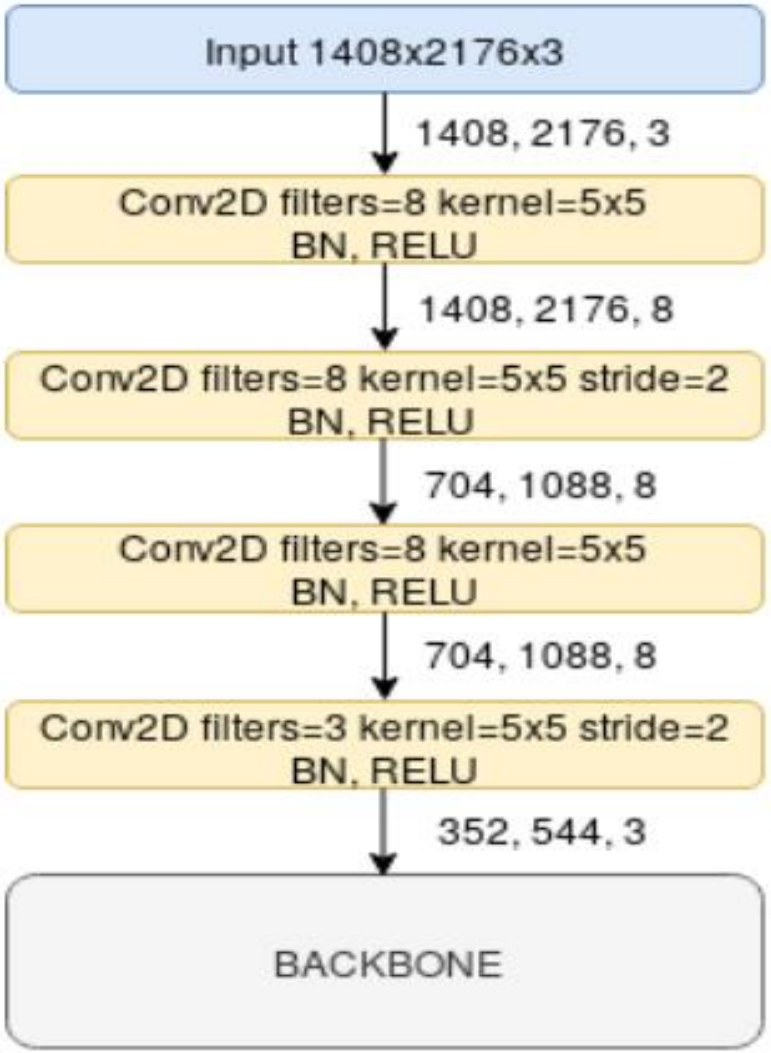
```
cls_probabilities = np.sort(mask_probabilities.reshape(4, -1), axis=1)
cls_probabilities = np.mean(cls_probabilities[:,-17500:], axis=1)
```

- Max probability를 positive prediction으로 사용
- 이미지당 적어도 1개 이상의 객체가 있으므로, max probability를 이용하여, 적어도 1개 이상의 물체를 탐지하도록 하였다.

```
cls_probabilities[np.argmax(cls_probabilities)] = 1
```

2. Solution

(1) 2nd Solution Summary



2nd soltuion

- 이미지를 resize (compress) 할 때 conv layer를 사용

출처: Andrey Kiryasov, “https://www.kaggle.com/c/understanding_cloud_organization/discussion/118255”

2. Solution

(1) 2nd Solution Summary

2nd soltuion

	Backbone		Input image dim		Image preprocess	
	efficientnetb1	seresnext50	1408 x2176	1280 x 1920	None	CLAHE
m1	✓		✓		✓	
m2	✓		✓			✓
m3	✓			✓	✓	
m4	✓			✓		✓
m5		✓	✓		✓	
m6		✓	✓			✓
m7		✓		✓	✓	
m8		✓		✓		✓

출처: Andrey Kiryasov, “https://www.kaggle.com/c/understanding_cloud_organization/discussion/118255”

2. Solution

(1) 2nd Solution Summary

2nd soltuion

Triplet scheme (top_score_threshold, min_contour_area, bottom_score_threshold)

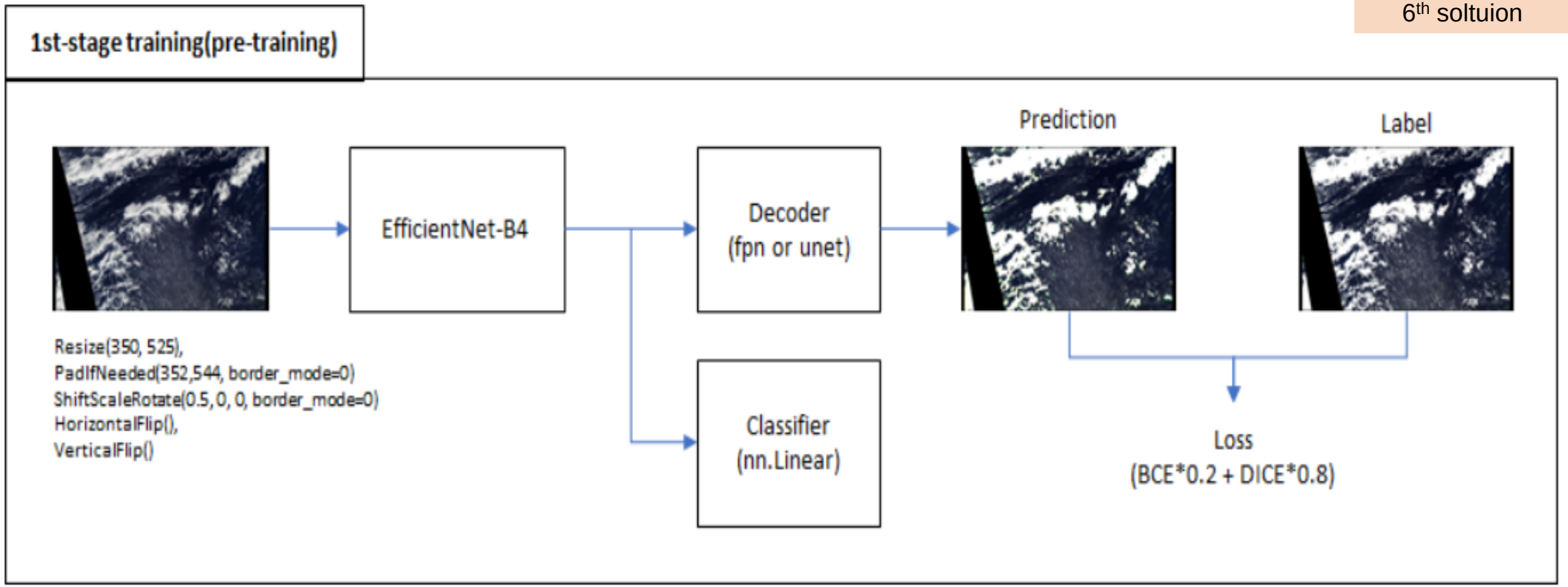
```
classification_mask = predicted > top_score_threshold
mask = predicted.copy()
mask[classification_mask.sum(axis=(1,2,3)) < min_contour_area, :,:,:] =
np.zeros_like(predicted[0])
mask = mask > bot_score_threshold
return mask
```

출처: Andrey Kiryasov, “https://www.kaggle.com/c/understanding_cloud_organization/discussion/118255”

2. Solution

(1) 6th Solution Summary

6th soltuion

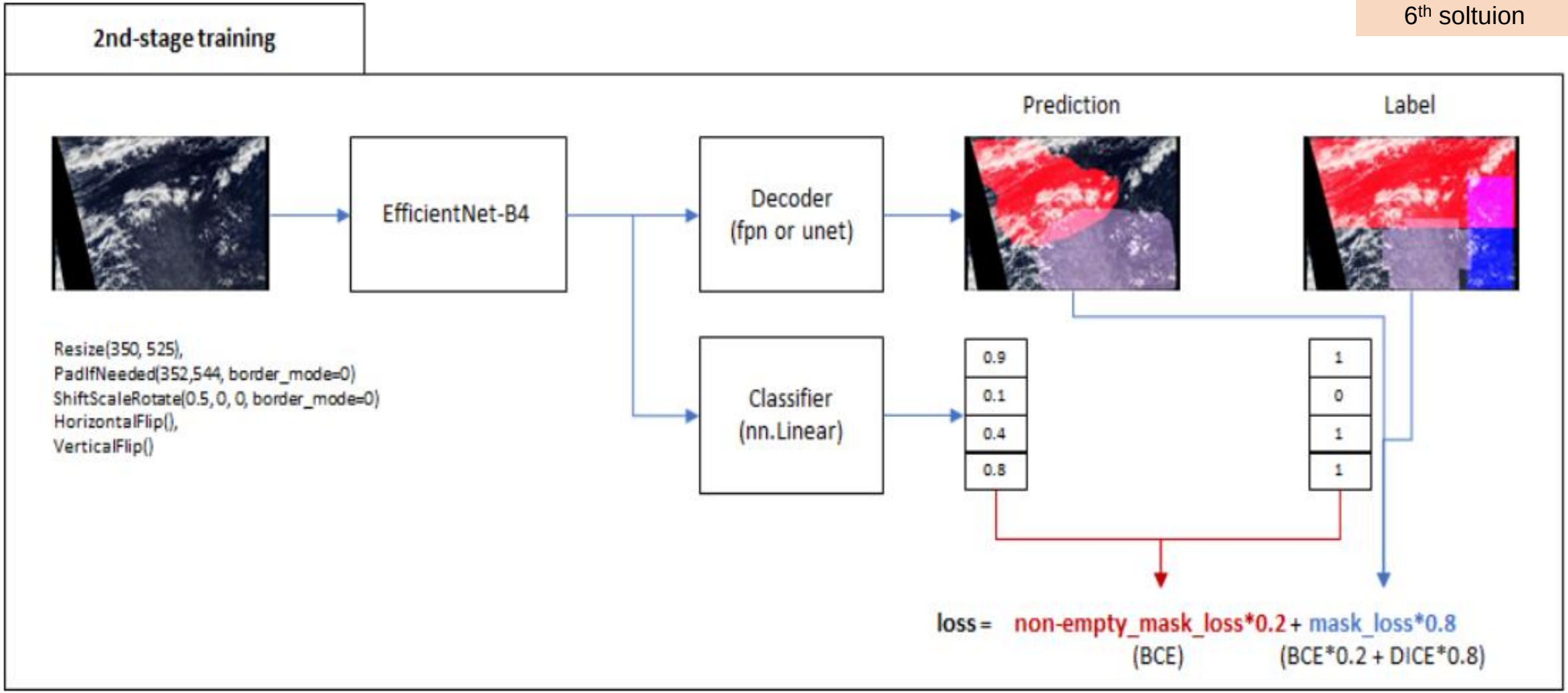


출처: Limerobot, “https://www.kaggle.com/c/understanding_cloud_organization/discussion/118017”

2. Solution

(1) 6th Solution Summary

6th soltuion



출처: Limerobot, “https://www.kaggle.com/c/understanding_cloud_organization/discussion/118017”