**Overview**

Congrats on achieving a prize winning rank in the SpaceNet-8, Flood Detection Using Multiclass Segmentation challenge. As part of your final submission and in order to receive payment for this match, please complete the following document.

1. Introduction

Tell us a bit about yourself, and why you have decided to participate in the contest.

* Name: KARI-AI   
  (members: Han Oh(captain), Hoonhee Lee, Dongoo Lee, Sungsik Huh, and Yeonju Choi)
* Handle: ohhan777, kari\_sat, ldg810, sshuh, anna\_choi (in the order of the above names)
* Placement you achieved: TBA
* About you: We work for the Korean space agency, Korea Aerospace Research Institute (KARI), and met in an academic club related to AI research. Each member’s specialty is diverse, including computer vision, aerospace, deep space exploration, physics, drones, etc.
* Why you participated in the challenge: We have known the SpaceNet dataset since SpaceNet Challenge 1 as a representative dataset for satellite image analysis. KARI operates several satellites called KOMPSAT, and we are also preparing AI datasets for KOMPSAT images. We wanted to know more about the newly released SpaceNet dataset, so we participated in this contest for the first time.

2. Solution Development

How did you solve the problem? What approaches did you try and what choices did you

make, and why? Also, what alternative approaches did you consider?

* We started with the baseline algorithm, and it helped us a lot to understand the dataset. We tried to apply a new algorithm there, but it was disadvantageous in terms of multi-GPU utilization and visualization, so we started implementing algorithms one by one in our semantic segmentation framework.
* When a building segmentation algorithm was implemented alone using the HRNet+OCR model, the performance began to exceed 60 points. (We still used the results of the baseline for the rest.)
* Then, a separate road speed segmentation algorithm was implemented using the HRNet+OCR model. By transforming the road speed labels, we defined eight channels (0-6 for road speed and 7 for background) and applied softmax+Cross Entropy loss, which reached about 65 points.
* We have integrated building and road speed segmentation algorithms into one model that shares a backbone (HRNet). The overall training time was shorter than that implemented separately, but the performance was almost identical.
* We tried to use one of the recent change detection algorithms (e.g., ChangeFormer, ChangeStar, DDPM-CD, Tiny\_model\_4\_CD) in the remote sensing community, but they did not help improve performance by themselves. Among them, we increased the parameter size of the Tiny\_model\_4\_CD algorithm, and it showed good performance, exceeding 70 points, but the memory requirement was too large to use.
* So, we implemented the flood detection algorithm using the Siamese HRNet+OCR model and scored over 70 points.
* Building, road speed, and flood segmentation algorithms were all integrated into one network model to reduce training time, and extensive augmentation was applied to improve performance.
* We spent considerable time on post-processing, but the effect was insignificant, so we only performed parameter tuning.
* In order to reduce the false positive of flood detection, a threshold was applied in a more conservative direction.
* The loss function initially applied the cross entropy loss with class weight but later applied the RMI loss, resulting in better performance.

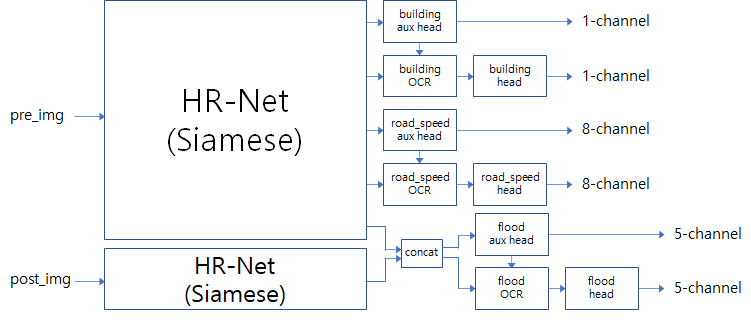
3. Final Approach

Please provide a bulleted description of your final approach. What

ideas/decisions/features have been found to be the most important for your solution

performance.

* The network structure that performs segmentation of buildings, roads, and floods at once is as follows.



* No other external data was used as training data; instead, extensive augmentation was used to increase the diversity of the data. The following is the list of transforms for albumentations.

T = [ A.MotionBlur(blur\_limit=(3,7), p=0.18),

A.CLAHE(p=0.25),

A.GaussNoise(var\_limit=(10.0,30.0), per\_channel=True, mean=0.0, p=0.18),

A.RGBShift(r\_shift\_limit=(-13,13), g\_shift\_limit=(-15,60), b\_shift\_limit=(-13,13),  
 p=0.18),

A.HueSaturationValue(hue\_shift\_limit=(-10,10), sat\_shift\_limit=(-10,10),

val\_shift\_limit=(-10,10), p=0.23),

A.RandomBrightnessContrast(p=0.30),

A.RandomGamma(p=0.15)]

* We used the BCE loss for building segmentation and the RMI loss for road speed and flood segmentation. There are two channel outputs for each task, and when calculating a loss, the weighted sum was applied by applying 0.4 for the aux channel and 1.0 for the OCR channel.
* The total loss is defined as follows.  
  total\_loss = 0.6 \* building\_loss + 0.4 \* roadspeed\_loss + 0.4 \* flood\_loss
* Through data analysis, we have identified that false positives for flood detection significantly impact the score and set flood detection to be more conservative. This is done in two ways. First, when determining flood, flood is defined only when the softmax value for the flood pixel was greater than the flood threshold value (flood\_thresh). Second, when flooded buildings and roads occur at a low rate in the image, they were considered false detection, and all of them were treated as non-flooded. (This is a heuristic approach, and the use is optional)
* We combined the building and road segmentation results and the building and road segmentation results obtained from flood detection, which were sometimes inconsistent, through OR operation to ensure consistency.
* Although our model is an integrated model, loading the best weights of building, road, and flood each showed slightly better performance than loading the best weight of the average.

4. Open Source Resources, Frameworks and Libraries

Please specify the name of the open source resource, the URL to where it can be found, and it’s license type:

* YOLOv5, [https://github.com/ultralytics/yolov5](﻿https://github.com/ultralytics/yolov5), and GPL3.0
* HRNet-Sematic Segmentation, [https://github.com/HRNet/HRNet-Semantic-Segmentation](﻿https://github.com/HRNet/HRNet-Semantic-Segmentation), and MIT
* NVIDIA Semantic Segmentation, <https://github.com/NVIDIA/semantic-segmentation>, and BSD 3-Clause
* gdal, <https://anaconda.org/conda-forge/gdal>, and MIT
* albumentations, <https://anaconda.org/conda-forge/albumentations>, and MIT
* W&B, <https://anaconda.org/conda-forge/wandb>, and MIT
* PyTorch, <https://pytorch.org>, and BSD

5. Potential Algorithm Improvements

Please specify any potential improvements that can be made to the algorithm:

* For some images, pre-image and post-image did not match significantly. It seems necessary to implement an additional image registration algorithm for this case.
* Currently, the algorithm is designed to operate under the premise that labels are accurate, but if there is a code to detect wrong labels, the performance will be better.
* If external data can be used, the performance will be more robust.

6. Algorithm Limitations

Please specify any potential limitations with the algorithm:

* It is cumbersome to use external data with an integrated network model.
* It relatively requires a lot of GPU memory

7. Deployment Guide

Please provide the exact steps required to build and deploy the code:

1. Step 1 -
2. Step 2 -
3. Step 3 -

8. Final Verification

Please provide instructions that explain how to train the algorithm and have it execute

against sample data:

1. Step 1 -
2. Step 2 -
3. Step 3 -

9. Feedback

Please provide feedback on the following - what worked, and what could have been

done better or differently?

* Problem Statement - Even though it was the first time participating, the problem statement was detailed so that it could be understood well, and other details could be grasped through the provided baseline algorithm.
* Data – We wish there were more training data.
* Contest - We are generally satisfied. It was beneficial to receive AWS credit.
* Scoring – We liked it because the evaluation speed was fast enough, and the metrics were reasonable.