A Solution to the ChaBuD Forest Fire Detection

Syusuke Yasui[1] and Hiroshi Yokoya[2]

[1] syuchimu@gmail.com
[2] hyokoya@gmail.com

Abstract. In this report, we explain our solution to the ChaBuD forest fire detection competition. Our model architectures are U-Net based segmentation models with post- and pre-fire images or post-fire image only as input. In addition to the provided train dataset, we utilize external Sentinel-2 images from the web, and trained our models with them. We suggest some attempts for further improving accuracy; test data augmentation, optimizing loss function, model ensembles, external data, etc.

1 Introduction

The ChaBuD challenge competition is hosted by Hagging Face [1]. The purpose of this competition is to explore methods for detecting forest fire by using Sentinel-2 satellite imageries and to support the assessment and management of forest fire. In the competition, participants are required to make a machine-learning model to predict the location of the forest fire from a pair of satellite images, one before the fire and the other after the fire. The images are taken by the Sentinel-2 satellite, containing 12 spectral bands, while removing the geographical information. The validation metric of the competition is the Intersection over Union (IoU) of the predicted area and the actual area of the forest fire.

Train dataset is composed of 534 sets of pre- and post-fire images, where each image has (512, 512) size and the 12 spectral bands. Some samples do not contain the pre-fire image, and some have partly cut-off area in the pre-fire image. Binary ground-truth labels, indicating the true forest fire area, are also provided. Usually, Sentinel-2 images have an image size of an order of 10000 x 10000. Thus, the dataset are supposed to be preprocessed by cropping the images into (512, 512) size patches. While the original Sentinel-2 image have different resolutions for the 12 spectral bands [2], the provided dataset images also seem to be preprocessed by up-sampling.

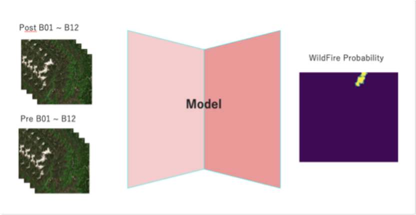
The private test dataset is composed of 68 sets of pre- and post-fire images, whose format is the same as those in the train dataset. Some pre-fire images have partly cut-off area, but there is no missing pre-fire image.

2 Preprocessing

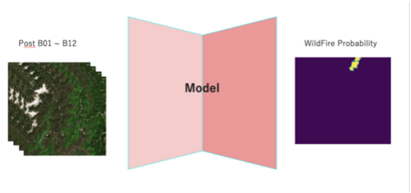
For our preprocess for the satellite images, we examined a static approach and a dynamic approach. The static approach do not change the pixel values but only convert the type of pixel values from integer to float, or clip the pixel values to a fixed range which is determined by checking the dynamic range which contribute to the forest fire area. The dynamic approach normalizes the images by using the minimum and maximum values (or 99, 99.5, 99.9 percentiles) in each channel and each image. In our analysis, we found that the static approach without clipping brings better results than the dynamic approach. By analyzing and visualizing the predicted area, we found that statistical features for the forest fire area are disturbed when the forest fire regions are large. Thus by dynamically normalizing the images, it becomes difficult for the model to learn the features of the forest fire area.

3 Model

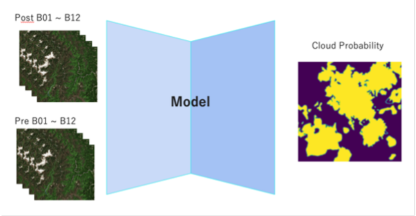
We considered three kinds of models for different purpose. The first two models are for the forest fire detection, and the last one is for cloud detection. One of the forest fire detection model is the Change Model which predicts the forest fire area by using the pre- and post-fire images. The other one is the Post Model which predicts the forest fire area by using the post-fire image only. The Cloud Model predicts the cloud area of the input image. It is used to avoid over-detections of the forest fire area, because of the obstruction by cloud.



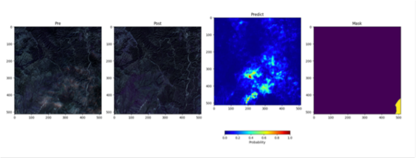
(a) Change model.



(b) Post model.



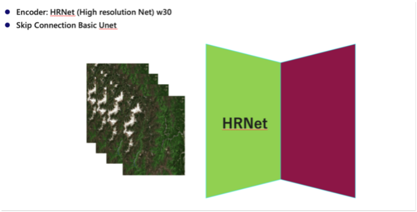
(a) Cloud model.



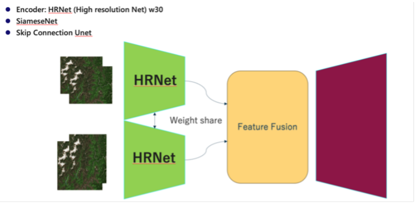
(b) Cloud model prediction.

4 Neural Net Architecture

To estimate the forest fire area, we use a probability-based segmentation model. Our model is based on the U-Net model [3] which is the standard model for segmentation tasks in the computer vision and remote sensing fields. We employ the High-Resolution Net (HRNet) [4] for the encoder part, which has a high-resolution feature map to detect small objects. In our experiment, the **w30** version of the HRNet shows the best performance, but the larger-size models, such as **w32**, **w40**, and **w48** do not improve the performance. For the Post Model and the Cloud Model, the input to the model is 12 channels indicating the Sentinel-2 multi-spectral bands. For the Change Model, the input to the model is the concatenation of the pre- and post-fire images, thus the 24 channels in total. We also examined the SiameseNet model [5] for the Change Model. The pre- and post-fire images (12 channels each) are fed into the separate backbone (HRNet) of the SiameseNet model, and the stacked feature map is fed into the decoder part. The decoder part is composed of the up-sampling layers with channel sizes of $(16, 16, 32, 48, 64)$. Larger channel sizes do not improve the performance.


Encoder: HRNet (High resolution Net) w30
Skip Connection Basic Unet
HRNet

(a) HRNet model.



(b) SiameseNet model.

5 Train

For the training of the models, we employ the augmentation methods of flips (horizontal, vetical), transpose and mix-up [6], because the rotation and scale transformation yield some artifacts between pixels which affect a delicate task, such as the detection of small objects in the satellite images. The reason of using mix-up is that we found that by adopting the mix-up with the probability of 0.5, a fast convergence in the model training is observed. We also tried some more augmentation methods, such as Gaussian noise, blur, brightness, contrast, and found that they contribute to stabilize the validation score.

In principle, the IoU loss is the best choice to optimize the IoU score. However, training becomes unstable by using the IoU loss due to the hypersensitivity to the positive ratio. Thus, a F_β loss with small β (such as 0.25) is better to reduce the false negative and to improve the IoU score in the end. The F_β loss is defined as $1 - F_\beta$, where $F_\beta = (1 + \beta^2)/(1/\text{precision} + \beta^2/\text{recall})$. To more stabilize, we utilize the hybrid of the F_β loss and the binary cross-entropy loss.

As a validation method of the models, we consider the cross-validation IoU score for our original 4 folds which are different from those provided by the competition host.

For the treatment of the defects of pre-fire images, just removing such dataset from the training data is slightly better than substituting the pre-fire image with the post-fire image.

For the training of the Cloud Model, we use the public dataset [7] which consists of the almost 12,000 Sentinel-2 images and the annotation mask on them. The satellite images in the dataset have four bands (B02, B03, B04 and B08), which are also in the images in the competition dataset. Thus the trained model can be applied for the competition dataset.

6 External Data

We include the external data for the training, namely the Sentinel-2 satellite images of Tokyo, Nagoya both in Japan, Arizona, Nevada, Oregon and Idaho in the US. We download 10 scenes for each area from sentinel hub [8]. The reason is that we find the inference results for the external California dataset has a lot of false positives and thus the model has to be trained with more negative samples.

However, it does not improve the convergence periods by simply adding negative samples. To solve it, we propose a mix random pair sampling method, that is randomly sampling the patches from the external sceneries, and then blending with the positive samples to fed into the training model. This method works well with external data, but not improve the score with the California dataset only. Because of the increase of the number of negative samples, upsampling of the positive samples by 4 to 8 times is also effective to stabilize the validation score.

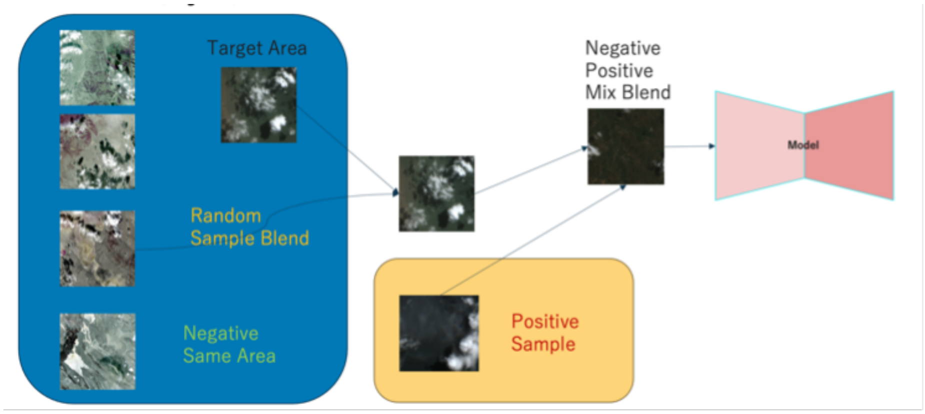
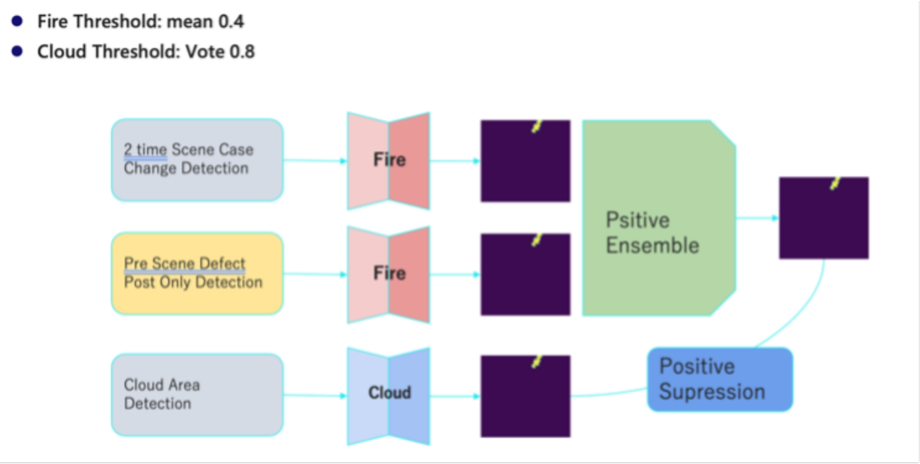


Fig. 4: HRNet model.

The threshold value optimization is performed by using the validation score. We employ the Powell method [9], and found that the averaged optimized threshold per images is within 0.3 to 0.4. By checking that 0.4 is better than 0.3, we choose 0.4 as the threshold value.

7 Inference

Our inference procedure utilizes the Change Model, the Post Model, and the Cloud Model. The area where the average score of the Change Model and Post Model is above the threshold are considered as the forest fire candidates. Then, the area which is also positive by the Cloud Model are removed from the forest fire candidates.



- Fire Threshold: mean 0.4
- Cloud Threshold: Vote 0.8

Fig. 5: Inference procedure.

Note that, here, the model is the ensemble of the 4-folds. To stabilize the inference output, we also utilize the Test Time Augmentation (TTA) [10] by flipping the image by (0, 90, 180, 270) degrees.

8 Data Cleaning

During our validation of the inference output, we found there are some images which are difficult to predict correctly. Indeed, by visualizing these images, some of them are hard to distinguish even by human eye, see Fig. 7.

By manually removing the images which are labeled as forest fire but cannot be found by eyes, or the images in which the forest fire can be found but not labeled in the mask, we made the Cleaning Data. For the later stage of our model trainings, we use this Cleaning data.

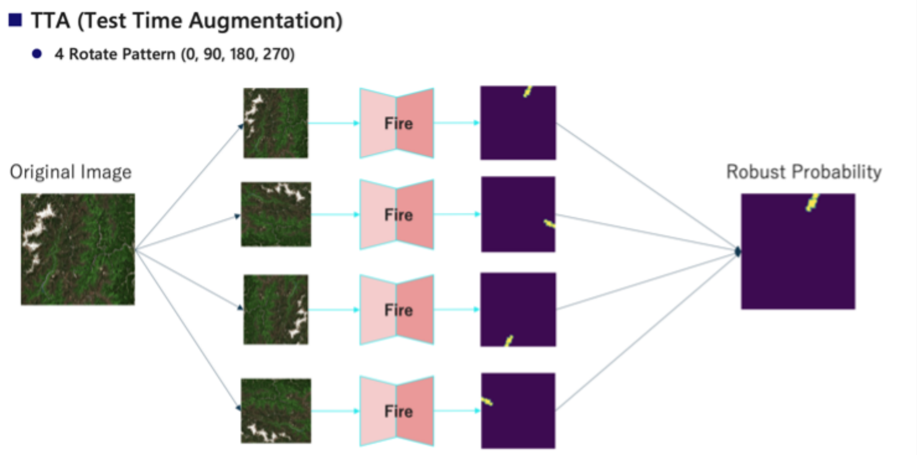

TTA (Test Time Augmentation)
4 Rotate Pattern (0, 90, 180, 270)
Original Image
Fire
Fire
Fire
Fire
Robust Probability

Fig. 6: Test Time Augmentation.

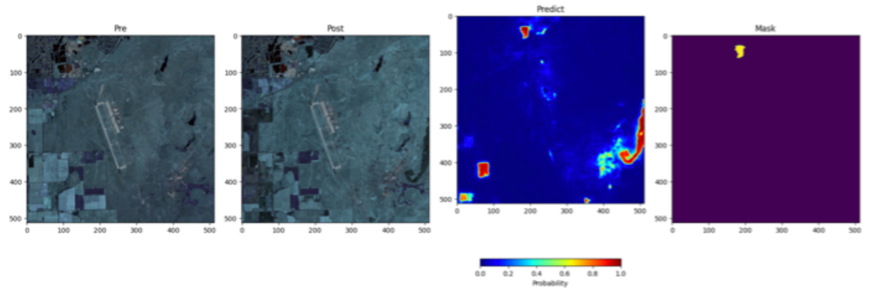


Fig. 7: Sample of hard-to-detect images by human eye.

9 Results

9.1 Cloud Model

The cross-validation IoU score of the Cloud Model is 0.888. This result is obtained for the ResNet**34** [11] backbone model with input size of 512. The optimizer is AdamW with learning ratio = 5e-4.

9.2 Change and Post Model

First, we compare our CV scores of single models. The results are shown in Table 1. For all the single models, Test Time Augmentation is used.

Model	Pre-fire	Data cleaning	External Data	CV score
Baseline	No	No	No	0.353
Post, Data Cleaning	No	Yes	No	0.327
Post, Data Cleaning, External	No	Yes	Yes	0.523
Post, Tokyo	No	No	Tokyo	0.488
Post, TNANO	No	No	TNANO	0.537
Change, TNANO	Yes	No	TNANO	0.523
Post, TNANOI	No	No	TNANOI	0.548
Change, TNANOI	Yes	No	TNANOI	0.517
Siamese	Yes	No	No	0.379
Siamese, TNANOI	Yes	No	TNANOI	0.522

Table 1: Status of single models, and the CV scores of them. TNANO represents Tokyo+Nagoya+Arizona+Nevada+Ohio external data, and TNANOI represents TNANO+Idaho.

Baseline Our baseline model is the Post Model without data cleaning and external data, whose CV score is 0.353.

Data Cleaning By cleaning the data, the CV score becomes 0.327.

External Data For the improvement of the CV score, the use of external data is effective. The highest score is obtained for the Post Model with Tokyo, Nagoya, Arizona, Nevada, Ohio, and Idaho external data, whose CV score is 0.548.

Change Model It was rather surprizing that the Change Model does not improve the CV score, as compared with the Post model, even though the increased input images.

SiameseNet Model The SiameseNet model is the Change Model with the Siamese network [5]. It improves the CV score from 0.353 to 0.379 without external data, but has little improvement with external data, 0.517 to 0.522.

9.3 Ensemble

To improve and stabilize the inference of our models, we ensemble the prediction of several models. The ensemble is performed simply by taking the average of the prediction of the models. We made the submission results by these ensemble models, and the results are partly shown in Table 2. We note that our CV folds are different from the folds provided by the competition. Therefore, our Public LB scores are high due to the leakage of the test dataset for the public LB.

Models	Public LB	Private LB
Baseline	-	0.717
Baseline + Cloud Model	0.815	0.628
Baseline + Post, TNANO + Cloud Model	0.815	0.641
same as above but thr=0.4	0.818	0.591
Data Cleaning + External	0.800	0.637
Post, TNANOI + Change, TNANOI with upsample16	0.804	0.673
Baseline + Siamese, TNANOI	0.815	0.637
Baseline + Siamese + Siamese, TNANOI	0.818	0.614

Table 2: Public and Private Leader Board scores of our submission models.

The effects of ensemble models and the choice of the final submission is discussed in the next section.

10 Analysis

First, we discuss the effects of the Cloud Model. The reason of introducing the Cloud Model is because the provided external California dataset has many cloud covered images. However, by visualizing the prediction of the cloud covered region on the private test dataset, we observed that the predicted cloud region and the forest fire region are sometimes overlapped. Thus, the Cloud Model can emerge the false-negative, even though it can reduce the false-positive. Because, the private test dataset has more cloud covered images than the train dataset, we anticipated that the Cloud Model is not effective for the private test dataset.

Next, we discuss the effects of ensemble of the models. We observed that ensemble of the models with data cleaning, external data with upsampling, which were effective to enhance our CV scores, however, do not contribute to the public LB scores. The ensemble models with SiameseNet models have better public LB scores, but the private LB scores turned out to be relatively low.

During the competition, it was seen that our CV scores and the public LB scores were not so correlated. (In addition, it turned out that the private LB scores are also not so correlated with the public LB scores.) Therefore, for the selection of the ensemble models or parameter choices, we visualized and checked the predicted masks. An example of the predicted masks for the private dataset is shown in Figure 8. These facts made our final model selection difficult. After opening the private LB, we realized that our baseline model has the highest private LB score among our submitted results.

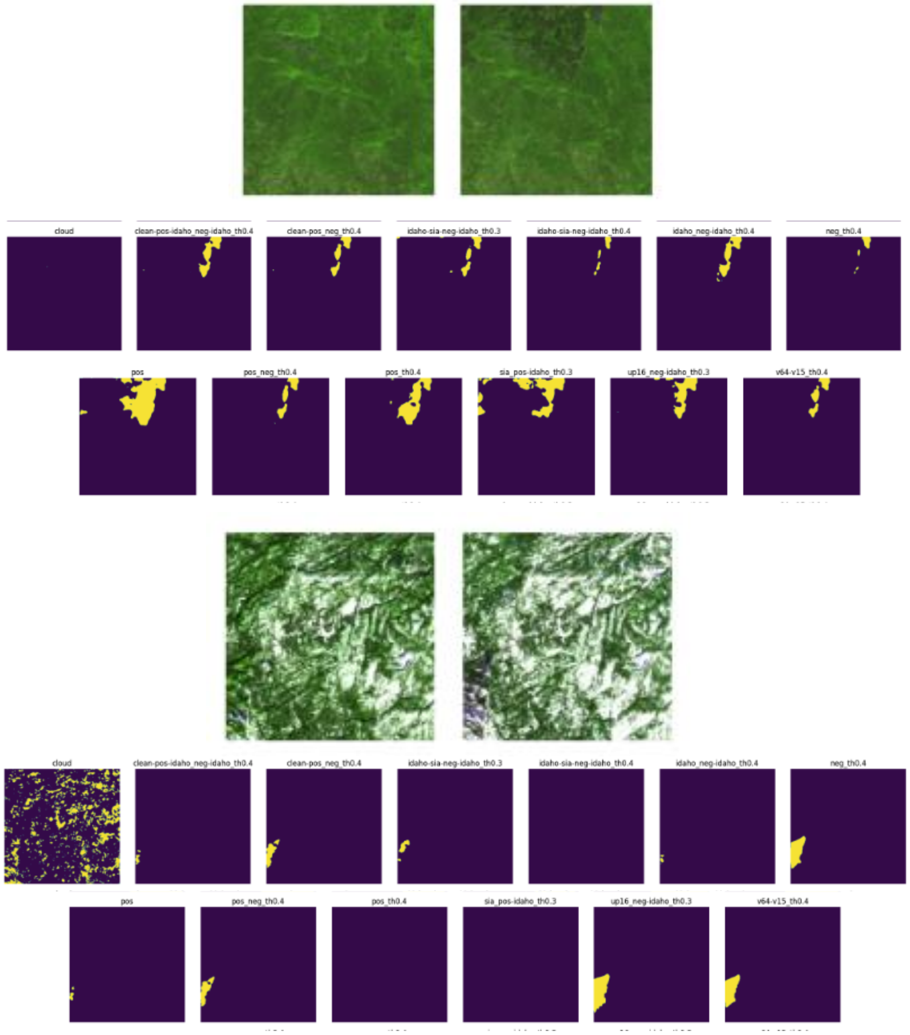


Fig. 8: Model prediction comparisons for private test data.

11 Implementation

Here, we list the other (hyper-) parameters, and our hardware environment.

- AdamW optimizer with learning rate = 1e-4, weight decay = 1e-4 to 1e-5.
- Cosine annealing scheduler with 1e-5.
- Batch size: 32
- Input size: 512
- Gradient clip: 15
- GPU: 1 (RTX3090, A100, or V100)
- Inference are tested on One RTX3090 GPU (memory 24GB)

12 Conclusion

In this report, we described our solution for the ChaBuD forest fire detection competition [1]. Our model architectures are U-Net based segmentation models with post- and pre-fire images or post-fire image only as input. We trained the model by using the provided train dataset with the combination of the F_β loss and the binary cross-entropy loss. We utilized the augmentation methods of flip, transpose, mix-up, and also data cleaning. We also included the external dataset which are downloaded from sentinel hub by using the mix random pair sampling method. The submission results are calculated by ensemble of the models. We consider the methods and ideas used in our studies are useful for generalization and stabilization of the model inference.

Code

Our code is available at https://github.com/syu-tan/ChaBuD-ECML-PKDD2023-solution.

References

1. Hagging Face competitions: https://huggingface.co/spaces/competitions/ChaBuD-ECML-PKDD2023.
2. Sentinel-2 MultiSpectral Instrument: https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-2-msi/msi-instrument.
3. Olaf Ronneberger, Philipp Fischer, Thomas Brox: ”U-Net: Convolutional Networks for Biomedical Image Segmentation”, https://arxiv.org/abs/1505.04597.
4. Jingdong Wang et al.: ”Deep High-Resolution Representation Learning for Visual Recognition”, https://arxiv.org/abs/1908.07919.
5. Vít Růžička, Stefano D’Aronco, Jan Dirk Wegner, Konrad Schindler: ”Deep Active Learning in Remote Sensing for data efficient Change Detection”, https://arxiv.org/abs/2008.11201.
6. H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, ”mixup: Beyond Empirical Risk Minimization”, https://arxiv.org/abs/1710.09412.

7. cloud-cover-detection: https://www.kaggle.com/datasets/hmendonca/cloud-cover-detection.
8. Sentinel Hub: https://www.sentinel-hub.com/.
9. Powell, M. J. D. (1964). "An efficient method for finding the minimum of a function of several variables without calculating derivatives". Computer Journal. 7 (2): 155-162. doi:10.1093/comjnl/7.2.155. hdl:10338.dmlcz/103029.
10. Divya Shanmugam, Davis Blalock, Guha Balakrishnan, John Guttag, "Better Aggregation in Test-Time Augmentation", https://arxiv.org/abs/2011.11156.
11. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun: "Deep Residual Learning for Image Recognition", https://arxiv.org/abs/1512.03385.