This CVPR workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Pseudo-label Generation and Various Data Augmentation for Semi-Supervised Hyperspectral Object Detection

Jun Yu[1], Liwen Zhang[1,†] , Shenshen Du[1], Hao Chang[1], Keda Lu[1],
Zhong Zhang[2], Ye Yu[3], Lei Wang[1], Qiang Ling[1]
[1]University of Science and Technology of China, [2]Hefei ZhanDa Intelligence Technology Co., Ltd,
[3]Hefei University of Technology
[1]{harryjun, wangl, qling}@ustc.edu.cn, [1]{zlw1113, changhaoustc}@mail.ustc.edu.cn,
[1]nibility163@163.com, [1]wujiekd666@gmail.com, [2]zhangzhong@zalend.com, [3]yuye@hfut.edu.cn

Abstract

Semi-supervised learning is a highly researched problem, but existing semi-supervised object detection frameworks are based on RGB images, and existing pre-trained models cannot be used for hyperspectral images. To overcome these difficulties, this paper first select fewer but suitable data augmentation methods to improve the accuracy of the supervised model based on the labeled training set, which is suitable for the characteristics of hyperspectral images. Next, in order to make full use of the unlabeled training set, we generate pseudo-labels with the model trained in the first stage and mix the obtained pseudo-labels with the labeled training set. Then, a large number of strong data augmentation methods are added to make the final model better. ***We achieve the SOTA, with an AP of 26.35, on the Semi-Supervised Hyperspectral Object Detection Challenge (SSHODC) in the CVPR 2022 Perception Beyond the Visible Spectrum Workshop, and win the first place in this Challenge.***

1. Introduction

In recent years, benefiting from the great success of deep learning [7, 11, 24], object detection [14, 19, 20] has already made great progress in the field of computer vision. By using a large amount of manually labeled data, the accuracy of object detection has been significantly improved. However, obtaining a large amount of manually labeled data, especially labeled data for object detection, requires precise localization and classification, which is labor-intensive. Hyperspectral images are images obtained by using hyperspectral camera, which have more number of bands and high resolution compared with ordinary RGB images, and contain rich spatial and spectral information. Based on these excellent properties, hyperspectral images will be able to better characterize the target if they are used as a training set for object detection. Hyperspectral images are more difficult to obtain compared to normal RGB images, and it is also more challenging to label hyperspectral images, which leads to a very small dataset on hyperspectral. Therefore, methods that utilize large amounts of unlabeled data to improve the performance of models are proposed in this situation. To be able to reduce the reliance on large amount of labeled data,

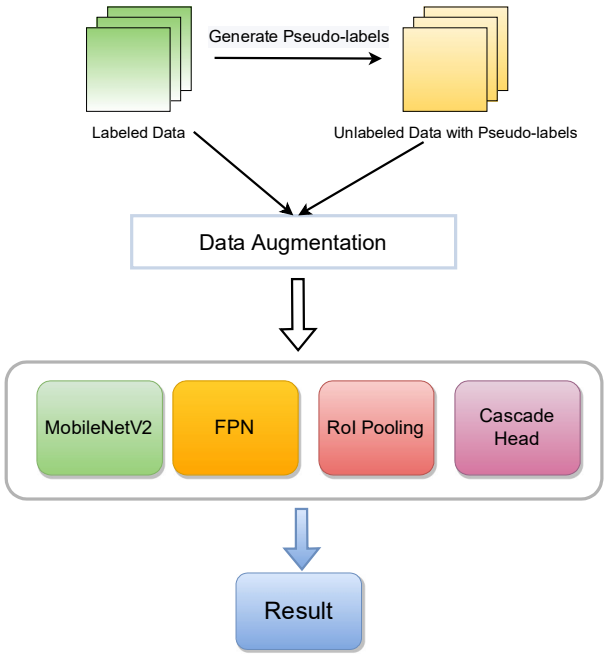

Figure 1. shows a brief version of our pseudo-label based object detection method.

† Corresponding author.

semi-supervised learning(SSL) [4] becomes one of the solutions to this challenge.

Semi-supervised learning can be trained on both labeled and unlabeled data, and in recent years a large number of methods have been proposed, which can be divided into four main categories [17]: Consistency based Learning [15], Pseudo-label based Learning [1, 8, 28], Generative Models [10, 21] and Graph based Learning [13]. Consistency based learning means that if a small perturbation is added to the unlabeled data points, the final prediction output should be similar. In pseudo-label based learning, the trained model which is trained on a few labeled data is used to make predictions on unlabeled data, which are filtered to generate pseudo-labels. These methods have significantly boosted the application of semi-supervised learning in the field of classification [28]. However, most of the semi-supervised learning focuses on the image classification domain, and semi-supervised object detection has been rarely addressed. The main reason is that object detection involves classification and regression of multiple classes on a single image.

Currently, semi-supervised object detection can be divided into two main types, Consistency based Learning [9] and Pseudo-label based Learning [26, 27], where the pseudo-label based semi-supervised object detection method is the method used in this paper. Fig. 1 shows a brief version of our pseudo-label based object detection. We use a portion of labeled data to train the model, and then predict the unlabeled data to get the classification and regression results on the images. The classification and regression frames with high confidence are then filtered using Non-Maximum Suppression(NMS) [16] and thresholding, and then retrained on the labeled and pseudo-labeled data after data augmentation.

Inspired by the application of semi-supervised learning in classification and object detection, Pseudo-label based Learning is applied to this hyperspectral semi-supervised object detection challenge. In this competition, there are 989 training sets, of which only 102 are labeled and the rest are unlabeled data. The challenge also provides 605 validation sets and 1296 test sets and the baseline of Faster r-cnn [20] based on MobileNetV2 [22]. We do not try more complex backbone due to the requirement of the number of parameters and computation of the model's backbone in the terms and conditions, and we still use MobileNetV2 as backbone.

For this case with less labeled data, we first exploit the labeled data to achieve a model that performs well on supervised learning. To make the detection results better, we use the better performing Cascade head [3] as the detector head, and then try a series of data augmentation and training tricks to improve the performance of the model, which enable us to achieve second place(50.81) on the validation set. The pseudo-label, which includes the category, confidence and the bounding box regression parameters, is used to filter the redundant bounding boxes by Soft-NMS [2], and the boxes with higher confidence are filtered by setting a threshold. The semi-supervised training is applied to the hyperspectral object detection, and the training is continued on the basis of the previous training. After attempting a series of data augmentation and training tricks, we finally win the 1st place in Semi-Supervised Hyperspectral Object Detection Challenge (SSHODC) held in the CVPR 2022 Perception Beyond the Visible Spectrum Workshop (PBVS).

2. Related Work

2.1. Object detection

Object detection is one of the most important applications in computer vision tasks and has made tremendous progress in the research community. Object detection can be classified into single-stage and two-stage approaches depending on whether Region Proposal Network(RPN) is used. Single-stage object detection [14, 19] methods produce classification and regression results directly, while two-stage methods use RPN networks to generate a series of RoIs, which are then classified and regressed separately. Faster r-cnn [20] is a classic of two-stage object detection methods, and many object detection networks have been developed based on it. For example, Cascade r-cnn [3], which uses a cascaded detector with incremental thresholds for each head, does not produce overfitting due to a sufficient number of proposals, and also solves the mismatch phenomenon [3]. Nevertheless, if we want to obtain a model with excellent generalization, we need a large amount of labeled data, which is obviously not very realistic.

2.2. Semi-supervised learning

In recent years, with the development of deep learning, the application of semi-supervised learning in the field of classification [28] has received more and more attention from researchers. Semi-supervised learning mainly includes two methods, Consistency regularization based methods [15, 18, 25] and Self-training-based methods [1, 8, 28], where the first method adds some small perturbations to the input and then minimizes the difference between the output predictions, thus training the model by constraining the data before and after the perturbation corresponding features to train the model. The second method first trains a model on labeled data, then uses the model to predict unlabeled data, then filters the predictions for labels with higher confidence, and finally trains on pseudo-labeled data.

2.3. Semi-supervised object detection

Semi-supervised object detection can be used to train models using large amounts of unlabeled data. There are

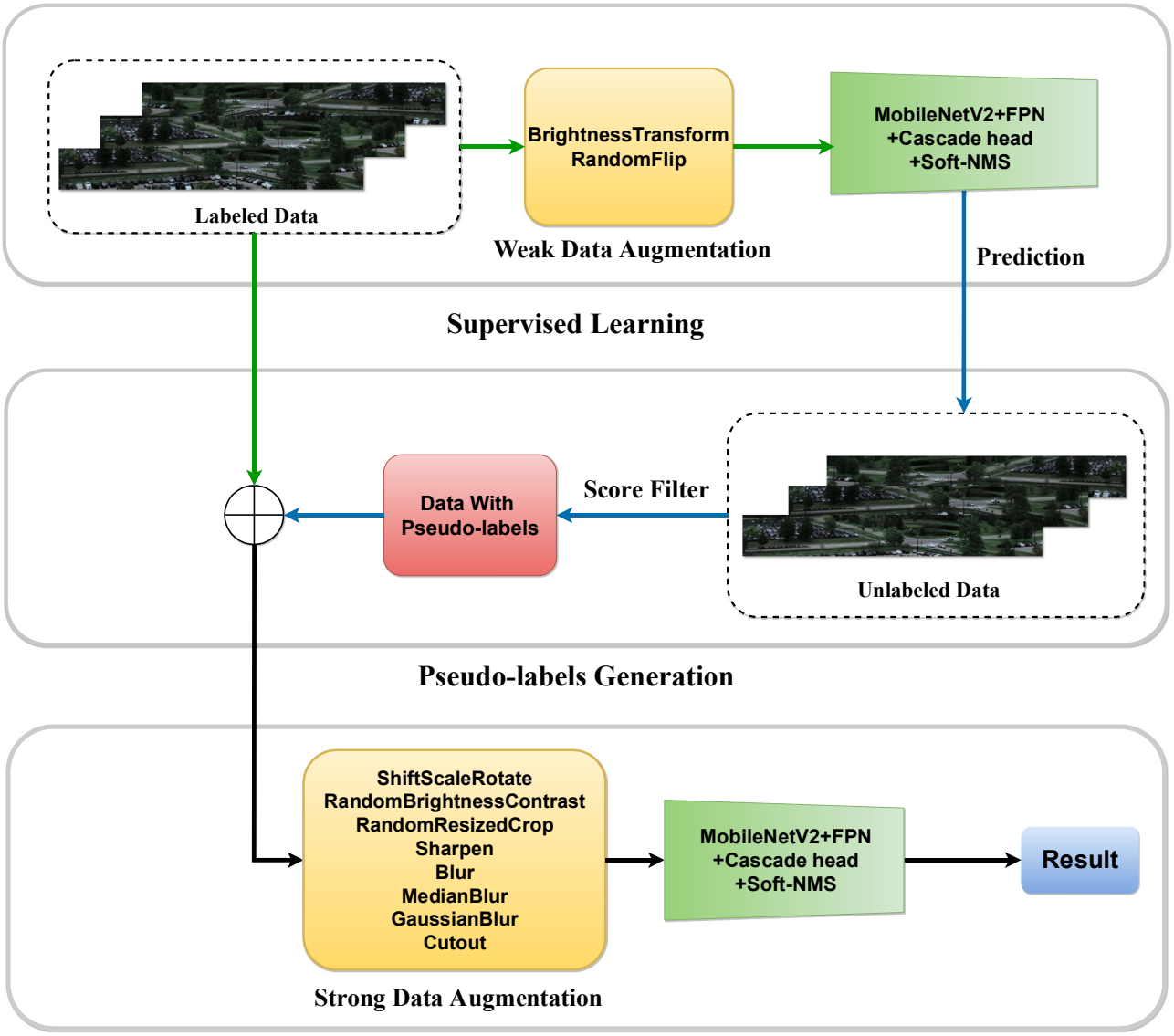

Figure 2. An overview of our solution.

two main approaches, Consistency based Learning [9] and Pseudo-label based Learning [26, 27]. The former uses two deep convolutional neural networks to learn the consistency between different perturbations (e.g. horizontal flip, different contrast, brightness, etc.) of the same unlabeled image, making full use of the information in unlabeled data. The latter approach borrows ideas from semi-supervised learning, but specifically for the object detection task, the generated pseudo-labels include categories and regression bounding boxes, which is more complex than the classification task. The reason is that the labeling of object detection is inherently more complex than classification. Finally the pseudo-labels are used to retrain the model.

3. Proposed Method

Fig. 2 illustrates the framework of our approach, which can be seen in three phases, the supervised learning phase, the pseudo-labels genetation phase and the semi-supervised learning phase. We first train a model using the existing labeled training set, and use this model to generate pseudo-labels by inference on the unlabeled training set. After generating the pseudo-labels, the pseudo-labelled images are put together with the original labeled images and then the last stage of training is performed to obtain the final re-

sults. Different data augmentation strategies are used in these two phases of training, and the key data augmentation methods used and the details of generating pseudo-labels are described below.

3.1. Data augmentation

Since the dataset is based on hyperspectral data, many data augmentation methods based on RGB images do not work well, and we also find that it is better to use a combination of different data augmentation methods in the two phases of training than to use the same data augmentation method in the two phases of training. In the supervised training phase, due to the small amount of data with labels, directly using strong data augmentation methods will make the accuracy of the trained model poor, and using a weaker data augmentation method for training can instead make the accuracy of the model higher, which in turn can improve the quality of the obtained pseudo-labels. Unlike the supervised training phase, in the semi-supervise learining phase, two problems must be considered.

1. Even if we have selected the pseudo-labels according to the higher score filter, the pseudo-labels may not be completely accurate due to the lack of accuracy of the model itself.

2. Since there are far more unlabeled data than labeled data, after mixing the unlabeled data set with the labeled data set, the percentage of unlabeled data set is very high, which makes the overall accuracy of the labels decrease. Therefore, in the second stage of training, we introduce some new data augmentation methods to improve the generalization and accuracy of the model, in addition to the data augmentation methods that proved to be very effective in the first stage. The following are some of the data augmentation methods that we have chosen for this task. And as shown in Tab. 1, we design experiments based on different combinations of data augmentation methods and demonstrate that all of these methods are effective.

3.1.1 BrightnessTransform

This method is used in both the first and second stage of training. On the one hand, because the dataset is selected from hyperspectral image values at different times of the day over a three-day period, changing the brightness can simulate the weather conditions at different times of the day to some extent; on the other hand, this transformation is very suitable for the characteristics of hyperspectral images.

3.1.2 Resize

Resize the image is a common data augmentation method in object detection. Unlike the conventional resize operation, we count the image ratio of the dataset and adjust the width and height to [(1600,189), (1600,188)], and randomly select the above mentioned set of width-height combinations and keep the aspect ratio constant during training. The reason for choosing to keep the aspect ratio constant here is that the class of the dataset contains vehicle, bus and bikes. These vehicles have their relatively fixed rigid structures, and if the aspect ratio is changed during Resize, it may lead to distortion of the objects in the images and thus decrease the final accuracy.

3.1.3 Spatial transformation

It can be found that introducing random combinations of spatial transformations during the second stage of training can improve the generalization of the model. We try to include ShiftScaleRotate, RandomResizedCrop, etc. in the second stage. Since the best model trained in the first stage is selected as pre-trained model in the second stage training, random combination according to probability by these methods can increase the sample diversity and avoid overfitting on the dataset.

3.1.4 Cutout

Cutout [6] is randomly cutting out part of the sample and filling it with 0 pixel values. Cutout enables the CNN to use the global information of the whole image instead of the local information composed of some small features. After experiments, selecting too large regions to cut off will lead to accuracy decrease, but randomly selecting smaller regions to cut according to the image aspect ratio can effectively improve the accuracy of the model.

3.2. Pseudo-labels Generation

In semi-supervised object detection tasks, pseudo-labeling is a common approach. However, the quality of pseudo-labels depend on the accuracy of the supervised model and the selection of the score filter. Too low a score will lead to a significant increase in false labels, but too high a score will miss some of the originally detected correct labels on the one hand, and reduce the number of available labels on the other. In our task, we select a threshold value of 0.99 for the score, which is found to be the better after experiments. As shown in Tab. 2, mix the pseudo-labels obtained when the Score-filter is taken to 0.99 with the original data with labels can achieve the best results in the second stage of training.

4. Experiments

4.1. Dataset

Unlike normal RGB images, the SSHODC dataset uses images generated by a hyperspectral camera at a spatial resolution of 189-212$\times$1600 pixels with 371 spectral bands, and has been downsampled to 51 bands. Each image in

Method	BrightnessTransform	Resize	Spatial transformation	Cutout	AP(val)	AP(Test)
Faster r-cnn	-	-	-	-	32.21	-
Faster r-cnn	✓	-	-	-	35.07	-
Cascade r-cnn	-	-	-	-	34.12	-
Cascade r-cnn	✓	-	-	-	45.01	-
Cascade r-cnn	-	✓	-	-	37.9	-
Cascade r-cnn	-	-	✓	-	31.48	-
Cascade r-cnn	✓	✓	-	-	46.12	20.97
Cascade r-cnn + Pseudo-labels	✓	✓	-	-	50.81	-
Cascade r-cnn + Pseudo-labels	✓	✓	✓	-	-	24.85
Cascade r-cnn + Pseudo-labels	✓	✓	-	✓	-	23.46
Cascade r-cnn + Pseudo-labels	✓	✓	✓	✓	-	26.35

Table 1. We design experiments based on different combinations of data augmentation methods, where Cascade r-cnn + Pseudo-labels represent the second stage of training after adding pseudo-labels. Note that some methods cannot be tested on both the validation set and the test set because there is a limit on the number of test set submissions in the competition, and the submission of validation set results is prohibited after the test set is published.

Method	Score filter	AP(val)
Cascade r-cnn	None	46.12
Cascade r-cnn	0.98	39.91
Cascade r-cnn	0.985	43.91
Cascade r-cnn	**0.99**	**50.81**
Cascade r-cnn	0.9925	45.3
Cascade r-cnn	0.995	48.52

Table 2. The effect of choosing different score filters on AP.

First score filter	The second score filter	AP(val)
0.99	-	**50.81**
0.99	0.99	44.3
0.99	0.9925	43.91

Table 3. The first score filter refers to the score filter selected when the model obtained from supervised training is first inferred to generate the pseudo-labels, while the second score filter is the score filter threshold selected when the pseudo-labels are obtained again after the first semi-supervised learning phase.

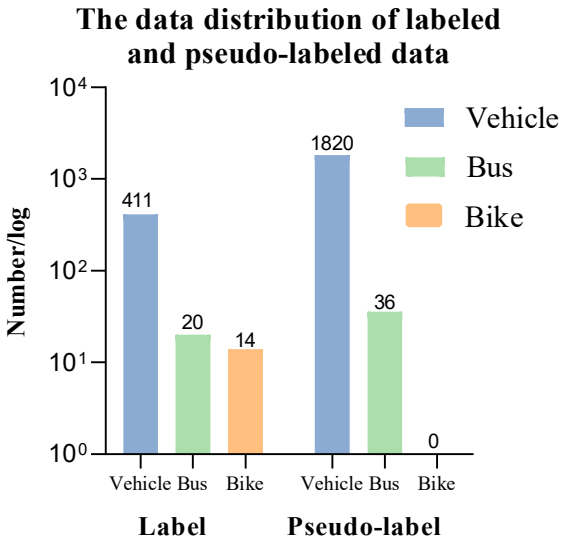

Figure 3. It shows the statistics of the number of categories contained in the labeled data and the number of categories of the generated pseudo-labels.

the dataset includes visible and near-infrared measurements captured over a three-day period at a fixed viewpoint. The dataset has a total of 989 images in the training set, 605 images in the validation set, and 1296 images in the test set. The training set is derived from data taken in the morning, and only 10% of the training set is labeled. The labeled data set has three categories: vehicle, bus, and bike. For each image, the dataset provides a processed image with 51 bands and a mask with the region of interest.

In our experiments, we use 102 labeled images as our training set for supervised training, followed by inference of the obtained model on the remaining 887 unlabeled images to generate pseudo-labels, and finally 989 hyperspectral images containing pseudo-labels and true labels as our training set for the second stage of training. As shown in Fig. 3, the number of each category in the real and generated labels is shown. It can be seen that the number of vehicles as a category is much higher than the number of buses and bicycles.

4.2. Implementation Details

Unlike the Faster r-cnn [20] provided by the competition organizers, we use Cascade r-cnn [3] equipped with FPN [12](Feature Pyramid Network) as our default detec-

Method	Soft-NMS	iou threshold	min score	AP(val)	AP(test)
Faster r-cnn	-	-	-	35.07	-
Faster r-cnn	✓	0.5	0.05	35.26	-
Cascade r-cnn	-	-	-	46.07	-
Cascade r-cnn	✓	0.5	0.05	46.12	20.97
Cascade r-cnn	✓	0.45	0.05	46.92	21.57
Cascade r-cnn	✓	0.4	0.05	46.54	-
Cascade r-cnn	✓	0.35	0.05	45.91	-
Cascade r-cnn	✓	0.3	0.05	45.23	-
Cascade r-cnn + Pseudo-labels	✓	0.45	0.05	-	25.98
Cascade r-cnn + Pseudo-labels	✓	0.45	0.01	-	**26.35**

Table 4. The effect of different parameter variations of Soft-NMS.

tion framework. As Tab. 1 shows, the accuracy of Cascade r-cnn is much higher than that of Faster r-cnn. Considering all submissions in the competition must be made by neural network backbone or traditional computer vision frameworks with a computation complexity equivalent or lesser than MobileNetv2 [22] (in terms of parameters and GFlops), we choose MobileNetv2 as our backbone. Our implementation and hyper-parameters are based on MMDetection [5]. Experiments are also conducted using MindSpore. Anchors with 5 scales and 3 aspect ratios are used. For the fully supervised phase of training and the semi-supervised phase of training, we select different training methods.

Fully supervised phase of training: In this phase, our model is trained on a V100, the batch size is set to 4. With SGD training, the learning rate is initialized to 0.01, the weight decay and the momentum are set to 0.0001 and 0.9. At the same time, we introduce warm up, where warm up iters is set to 1000 and warm up ratio is set to 0.1, and the learning rate is stepped up at the 10th, 20th, and 25th epochs, respectively.

Semi-supervised phase of training: In this phase, our model is trained on a V100, the batch size is set to 8. With SGD training, the learning rate is initialized to 0.02, the weight decay and the momentum are set to 0.0001 and 0.9. Other optimizer settings are the same as in the fully supervised training phase. Unlike the previous stage, in this one, we add many data augmentation strategies. According to our settings, there will be a 0.5 probability of ShiftScaleRotate operation on the image, with shift limit set to 0.0625; there will be a 0.2 probability of RandomBrightnessContrast, with both brightness limit and contrast limit at [0.1, 0.3]; at the same time, there is a 0.2 probability of RandomResizedCrop; 0.1 probability of Blur, MedianBlur and GaussianBlur are added to the image.

We also try to generate pseudo-labels repeatedly, i.e., using the model trained in the second stage to generate pseudo-labels again, but as shown in Tab. 3, this do not work well, not even as well as without pseudo-labels.

In the inference stage, Soft-NMS [2] is used, which has two parameters: iou threshold and min score. When the inferred score is less than min score, it will be filtered out directly. We experiment with the selection of min score and finally find that the selection of 0.01 work best. In addition, we use TTA to randomly adjust the image width and height to a set of [(1600,188), (1600,189)] to further improve the final accuracy.

4.3. Ablation Studies

In this section, the validity of our approach are verified and we also do a lot of experiments to verify that the hyper-parameters have chosen are well worked for this task.

Effects of Soft-NMS. We confirm the validity of Soft-NMS and find the most effective set of parameters, and the results are shown in Tab. 4. It can be seen that a smaller min score can retain some results with lower score but correct inference. The reason for this result is that, on the one hand, choosing a smaller min score can preserve some inference results with lower sorce but accurate, and on the other hand, due to the limitation of the model itself, some accurate inference results cannot be obtained with a high socre. so by adjusting to the appropriate parameters, the model's capability can be fully utilized to improve the final accuracy.

OHEM vs. RandomSampler. In two-stage object detection methods, the region proposals generated by region generation algorithms or networks are usually screened for positive and negative samples and scaled before being fed into the subsequent detection network for training. for training. We compare this method with RandomSampler, but find that the results are not as good as RandomSampler, probably because there are too few labeled data, which affects the results of Online Hard Example Mining [23]. The results are shown in Tab. 5.

Effects of different optimizer settings. For SGD optimizer, a proper learning rate and batch size are crucial. When the learning rate is set too small, the convergence pro-

Method	Sampler	AP(test)
Cascade r-cnn	OHEM	18.94
Cascade r-cnn	RandomSampler	20.97

Table 5. Results using OHEM with RandomSampler on the test set.

cess will become very slow. When the learning rate is set too large, the gradient may oscillate back and forth around the minimum value and may not even converge. In the semi-supervised training process, we tried different combinations of learning rate and batch size based on the use of warm up, and the results are shown in Tab. 6. In the second stage of training, a learning rate of 0.02 and a batch size of 8 can achieve the best results.

Effects of Cutout. We confirm the validity of Soft-NMS and find the most effective set of parameters, and the results are shown in Tab. 4. It can be seen that a smaller min score can retain some results with lower score but correct inference. The reason for this result is that, on the one hand, choosing a smaller min score can preserve some inference results with lower sorce but accurate, and on the other hand, due to the limitation of the model itself, some accurate inference results cannot be obtained with a high socre. so by adjusting to the appropriate parameters, the model's capability can be fully utilized to improve the final accuracy.

Effects of different backbone. Considering all submissions in the competition must be made by neural network backbone or traditional computer vision frameworks with a computation complexity equivalent or lesser than MobileNetv2 [22] (in terms of parameters and GFlops), we choose MobileNetv2 as our backbone. But in fact, using a more complex backbone can improve the accuracy. We try some backbones early in the competition and experiment without adding any other strategy. The results are shown in Tab. 7. If the backbone is not restricted, the final accuracy should be further improved.

Learning rate	Batch size	AP(val)
0.001	4	46.54
0.002	8	45.23
0.01	4	47.1
0.02	8	**50.81**
0.04	16	44.3

Table 6. The effect of using different combinations of learning rate and batch size. The framework used in the table are Cascade r-cnn, and the results are semi-supervised stage training.

5. Conclusion

Hyperspectral images are vastly different from RGB images. In our work, we consider the characteristics of hyperspectral images and select a series of effective data augmentation methods to improve the accuracy of the model. To make full use of the unlabeled training set, we infer and obtain pseudo-labels from the unlabeled training set using the model trained with the labeled training set, and mix the pseudo-labels with the ground truth labels for training. To improve the generalization and accuracy of the final model, we replace different combinations of data augmentation methods in the second stage of training. With these methods, we achieve an AP of 26.35 on the SSHODC test set, which is the SOTA for this dataset, proving that our method is very effective for semi-supervised hyperspectral object detection. Besides, **our method win the championship at the CVPR 2022 PBVS SSHODC.**

Method	Backbone	AP(val)
Faster r-cnn	MobileNetv2	32.21
Cascade r-cnn	MobileNetv2	34.12
Faster r-cnn	ResNet50	26.47
Cascade r-cnn	ResNet50	30.3
Faster r-cnn	ResNeSt50 [29]	36.82
Cascade r-cnn	ResNeSt50	41.91
Cascade r-cnn	ResNeSt101	47.56

Table 7. The effect of using different backbone on the experimental results.

6. Acknowledgements

This work is sponsored by CAAI-Huawei MindSpore Open Fund (CAAIXSJLJJ-2021-016B), Anhui Province Key Research and Development Program (202104a05020007) and USTC Research Funds of the Double First-Class Initiative (YD2350002001).

References

[1] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27, 2014. 2

[2] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 2, 6

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 2, 5

[4] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009. 2

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu,

Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[6] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 4

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[8] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5070–5079, 2019. 2

[9] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. *Advances in neural information processing systems*, 32, 2019. 2, 3

[10] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014. 2

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1

[12] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 5

[13] Bin Liu, Zhirong Wu, Han Hu, and Stephen Lin. Deep metric transfer for label propagation with limited annotated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2

[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1, 2

[15] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. 2

[16] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 850–855. IEEE, 2006. 2

[17] Yassine Ouali, Céline Hudelot, and Myriam Tami. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*, 2020. 2

[18] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12674–12684, 2020. 2

[19] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1, 2

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 1, 2, 5

[21] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 2

[22] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 2, 6, 7

[23] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016. 6

[24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[25] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 2

[26] Fang Wan, Pengxu Wei, Jianbin Jiao, Zhenjun Han, and Qixiang Ye. Min-entropy latent model for weakly supervised object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1306, 2018. 2, 3

[27] Keze Wang, Xiaopeng Yan, Dongyu Zhang, Lei Zhang, and Liang Lin. Towards human-machine cooperation: Self-supervised sample mining for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1605–1613, 2018. 2, 3

[28] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020. 2

[29] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020. 7