



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Ph.D. DISSERTATION

Consistency & Interpolation-based
Semi-supervised learning for
Object Detection

객체검출 알고리즘을 위한
일관성과 보간법 기반의 준지도 학습

BY

JISOO JEONG

AUGUST 2021

Intelligent Systems
Department of Transdisciplinary Studies
Graduate School of Convergence Science and Technology
SEOUL NATIONAL UNIVERSITY

Consistency & Interpolation-based Semi-supervised learning for Object Detection

객체검출 알고리즘을 위한
일관성과 보간법 기반의 준지도 학습

지도교수 곽 노 준

이 논문을 공학박사 학위논문으로 제출함

2021년 8월

서울대학교 대학원

융합과학부 지능형융합시스템전공

정 지 수

정지수의 공학박사 학위 논문을 인준함

2021년 8월

| | |
|--------|--------------|
| 위 원 장: | <u>이 교 구</u> |
| 부위원장: | <u>곽 노 준</u> |
| 위 원: | <u>이 원 중</u> |
| 위 원: | <u>서 봉 원</u> |
| 위 원: | <u>이 민 식</u> |

Abstract

Object detection, one of the main areas of computer vision researches, is a task that predicts where and what the objects are in an RGB image. While the object detection task requires a massive number of annotated samples to guarantee its performance, placing bounding boxes for every object in each sample is costly and time consuming. To alleviate this problem, Weakly-Supervised Learning and Semi-Supervised Learning methods have been proposed. However, they show large gaps from supervised learning in efficiency and require a lot of research. Especially in Semi-Supervised Learning, the deep learning-based learning methods are not yet applied to object detection.

In this dissertation, we have applied the latest deep learning-based Semi-Supervised Learning methods to object detection, which considers and solves the problems caused by applying the established Semi-Supervised Learning algorithms. Specifically, we have adopted Consistency Regularization (CR) and Interpolation Regularization (IR) Semi-Supervised Learning methods to object detection individually and combined them together for performance improvement. It is the first attempt to extend CR and IR to object detection problem which was only used in conventional semi-supervised classification problems.

First, we propose a novel Consistency-based Semi-Supervised Learning method for object Detection (CSD), which is a way of using consistency constraints to enhance detection performance by making full use of available unlabeled data. To be specific, the consistency constraint is applied not only for object classification but also for localization. We also propose Background Elimination (BE) to avoid the negative effect of the predominant backgrounds on the detection per-

formance. We evaluated the proposed CSD both in single-stage and two-stage detectors, and the results show the effectiveness of our method.

Second, we present a novel Interpolation-based Semi-Supervised Learning method for object Detection (ISD), which considers and solves the problems caused by applying conventional Interpolation Regularization (IR) directly to object detection. We divide the output of the model into two types according to the objectness scores of both original patches that are mixed in IR. Then, we apply a separate loss suitable for each type in an unsupervised manner. The proposed losses dramatically improve the performance of Semi-Supervised Learning as well as supervised learning.

Third, we introduce the method of combining CSD and ISD. In CSD, it requires an additional prediction for applying consistency regularization, and it allocates twice ($\times 2$) as much memory as conventional supervised learning. In ISD, in addition, two supplementary predictions are computed for applying interpolation regularization, and it takes three times ($\times 3$) as much memory as conventional training. Therefore, it requires three extra predictions to combine CSD and ISD. In our method, by applying shuffle the sample in mini-batch in CSD, we reduced the additional predictions from three to two, which can cut back the memory. Furthermore, combining two algorithms shows performance improvement.

keywords: Semi-supervised learning, Object detection, Consistency regularization, Interpolation regularization, Deep learning

student number: 2015-26109

Contents

| | |
|--|------------|
| Abstract | i |
| Contents | iii |
| List of Tables | vii |
| List of Figures | ix |
| 1 INTRODUCTION | 1 |
| 1.1 Problem Definition | 5 |
| 1.2 Motivation | 6 |
| 1.3 Challenges | 7 |
| 1.3.1 Structure | 7 |
| 1.3.2 Localization | 7 |
| 1.3.3 Background | 8 |
| 1.3.4 Memory | 8 |
| 1.4 Contributions | 9 |
| 1.4.1 Consistency-based Semi-Supervised Learning for ob- ject Detection (CSD) | 9 |

| | | |
|----------|--|-----------|
| 1.4.2 | Interpolation-based Semi-Supervised Learning for object Detection (ISD) | 10 |
| 1.4.3 | Combination of CSD with ISD | 10 |
| 1.5 | Outline | 11 |
| 2 | Related works | 12 |
| 2.1 | Semi-supervised learning | 13 |
| 2.1.1 | Self-Training | 13 |
| 2.1.2 | Consistency Regularization | 14 |
| 2.1.3 | Interpolation Regularization | 15 |
| 2.2 | Object detection | 19 |
| 2.2.1 | Dataset | 19 |
| 2.2.2 | Evaluation metric | 21 |
| 2.2.3 | Survey of Object Detection Algorithms | 23 |
| 3 | Consistency-based Semi-supervised learning for object Detection (CSD) | 42 |
| 3.1 | Introduction | 42 |
| 3.2 | Method | 44 |
| 3.2.1 | Consistency loss for classification | 45 |
| 3.2.2 | Consistency loss for localization | 46 |
| 3.2.3 | Overall loss for object detection | 47 |
| 3.2.4 | Application to two-stage detector | 48 |
| 3.2.5 | Background Elimination | 49 |
| 3.3 | Experiments | 50 |
| 3.3.1 | Implementation Detail | 52 |
| 3.3.2 | Ablation Study | 53 |
| 3.3.3 | Unlabeled data with different distribution (MSCOCO) | 55 |

| | | |
|----------|--|-----------|
| 3.3.4 | MSCOCO | 57 |
| 3.4 | Discussion | 58 |
| 3.4.1 | Consistency regularization with only labeled data | 58 |
| 3.4.2 | Single-stage detector vs. Two-stage detector: | 60 |
| 3.4.3 | Background Elimination: | 60 |
| 3.4.4 | Datasets | 61 |
| 3.4.5 | Self-training vs. Consistency regularization | 63 |
| 3.5 | Conclusion | 64 |
| 4 | Interpolation-based Semi-supervised learning for object Detection (ISD) | 65 |
| 4.1 | Introduction | 65 |
| 4.2 | Method | 68 |
| 4.2.1 | Type categorization. | 69 |
| 4.2.2 | Type I loss | 69 |
| 4.2.3 | Type II loss | 71 |
| 4.3 | Experiments | 74 |
| 4.3.1 | PASCAL VOC | 74 |
| 4.4 | Discussion | 77 |
| 4.4.1 | Ablation studies for Type-I and Type-II losses | 77 |
| 4.4.2 | Beta distribution | 78 |
| 4.4.3 | Training model size | 79 |
| 4.4.4 | Object detector | 80 |
| 4.5 | Conclusion | 80 |
| 5 | Combination of CSD and ISD | 82 |
| 5.1 | Method | 85 |

| | | |
|----------|---|------------|
| 5.2 | Experiments | 85 |
| 5.2.1 | PASCAL VOC | 85 |
| 5.2.2 | Unlabeled data with different distribution (MSCOCO) | 89 |
| 5.2.3 | MSCOCO | 90 |
| 5.3 | Discussion | 90 |
| 5.3.1 | CSD and ISD with only labeled data | 90 |
| 5.3.2 | Small labeled dataset | 91 |
| 5.3.3 | Training model size | 92 |
| 5.4 | Conclusion | 92 |
| 6 | Conclusion | 98 |
| 6.1 | Summary | 99 |
| 6.2 | Limitations | 100 |
| 6.3 | Future Directions | 101 |
| | Abstract (In Korean) | 116 |
| | 감사의 글 | 118 |

List of Tables

| | | |
|-----|---|----|
| 2.1 | Pascal VOC and MSCOCO Dataset | 19 |
| 2.2 | Pascal VOC and MSCOCO Classes | 20 |
| 2.3 | VOC2007+2012 training and VOC 2007 test result | 35 |
| 2.4 | Recall for objects in different size [49]. | 37 |
| 2.5 | Results on VOC2007 test dataset trained with VOD2007 small train dataset | 38 |
| 3.1 | Single-Stage Detection results for PASCAL VOC2007 test set. . . | 54 |
| 3.2 | Two-Stage Detection results for PASCAL VOC2007 test set. . . | 56 |
| 3.3 | Detection results on PASCAL VOC2007 test set. | 57 |
| 3.4 | Detection results for MS COCO test-dev set. | 58 |
| 3.5 | Detection results for PASCAL VOC2007 test set. | 59 |
| 3.6 | Effects of using Background Elimination (BE) on SSD300 per- formance. | 60 |
| 3.7 | Comparisons between self-training and consistency regulariza- tion based methods on PASCAL VOC2007 test set. | 62 |
| 4.1 | Detection results for PASCAL VOC2007 test set under the su- pervised training setting. | 75 |

| | | |
|-----|--|----|
| 4.2 | Detection results for PASCAL VOC2007 test set under the semi-supervised training setting. | 76 |
| 4.3 | Ablation study of Type-II losses on PASCAL VOC2007 test set. | 78 |
| 4.4 | Ablation study for α and each type in VOC07(L) + VOC12(U) training dataset and VOC07 testing dataset. | 79 |
| 5.1 | Detection results for PASCAL VOC2007 test set under the supervised training setting. | 87 |
| 5.2 | Detection results for PASCAL VOC2007 test set under the semi-supervised training setting. | 88 |
| 5.3 | Detection results for PASCAL VOC2007 test set. | 89 |
| 5.4 | Detection results for MS COCO test-dev set. | 90 |
| 5.5 | Detection results for PASCAL VOC 2007 set. | 91 |

List of Figures

| | | |
|------|---|----|
| 1.1 | Annotation types and times for three different tasks. | 2 |
| 1.2 | Different types of object detection settings | 3 |
| 1.3 | Semi-supervised Learning for Object Detection | 5 |
| 2.1 | The overall structure of Self-Training. | 13 |
| 2.2 | The overall structure of Consistency Regularization. | 14 |
| 2.3 | Example of Conventional training and Mixup training methods in binary classification problem | 16 |
| 2.4 | Overview of the images of Cutout, Cowout, Mixup, CutMix, and our CowMix | 17 |
| 2.5 | Interpolation Consistency Training | 18 |
| 2.6 | Detection results according to the threshold. | 21 |
| 2.7 | Recall vs. Precision graph | 22 |
| 2.8 | Two types of object detectors in pre-deep learning era. | 24 |
| 2.9 | deep learning based two types of object detectors | 26 |
| 2.10 | Example of classifier of object detection based on deep learning | 27 |
| 2.11 | Conventional SSD vs. the proposed Rainbow SSD (R-SSD). . . | 30 |
| 2.12 | Proposed methods of feature concatenation | 32 |
| 2.13 | Conventional SSD vs. the proposed Rainbow SSD (R-SSD). . . | 40 |

| | | |
|-----|--|----|
| 3.1 | difficult to establish a one-to-one correspondence | 43 |
| 3.2 | Overall structure of our proposed method for single stage detector. | 45 |
| 3.3 | Overall structure of our proposed method for two stage detector. | 48 |
| 4.1 | Mixed image created by random interpolation between images A and B | 66 |
| 4.2 | (a) Type-I : both patches are from object classes. (b) Type-II : one of the patches is from the object class. | 67 |
| 4.3 | Type-I Loss : both patches are from object classes | 70 |
| 4.4 | Type-II : one of the patches is from the object class | 71 |
| 5.1 | The mixed image $Mix_{\lambda}(A, \hat{A})$ of $A \in \mathcal{A}$ and its horizontal flipped version $\hat{A} \in \hat{\mathcal{A}}$ | 83 |
| 5.2 | Combination of ISD with CSD. | 84 |
| 5.3 | Qualitative results for the PASCAL VOC2007 test set using su- pervised SSD, semi-supervised CSD and CSD+ISD models in table 5.2. | 94 |
| 5.4 | Qualitative results for the PASCAL VOC2007 test set using su- pervised SSD, semi-supervised CSD and CSD+ISD models in table 5.2. | 95 |
| 5.5 | Qualitative results for the PASCAL VOC2007 test set using su- pervised SSD, semi-supervised CSD and CSD+ISD models in table 5.2. | 96 |
| 5.6 | Qualitative results for the PASCAL VOC2007 test set using su- pervised SSD, semi-supervised CSD and CSD+ISD models in table 5.2. | 97 |

Chapter 1

INTRODUCTION

Object detection, one of the main topics of computer vision research, is the task of predicting the position of objects and their classes in an RGB image [60, 57, 58, 50]. It can facilitate high-level scene understanding from an image consisting of digital numbers for each pixel. Therefore, it is widely employed in applications, such as license plate recognition, surveillance cameras, driving aid, and autonomous cars [79, 8, 26]. Since improving the accuracy and speed of object detection directly benefits downstream tasks mentioned above, this research is crucial, and various methods have been studied [50, 60, 72, 7].

Semi-Supervised Learning (SSL) is a method alleviating the inefficiencies associated with the data collection and annotation process, which lies between supervised learning and unsupervised learning in that both labeled and unlabeled data are used in the learning process [9, 55]. It can efficiently train a model from fewer labeled data using a large amount of unlabeled data [92, 9]. Accordingly, the significance of SSL has been studied extensively in the previous literature [94, 62, 32, 56]. These results suggest that SSL can be a useful approach when the amount of annotated data is insufficient.

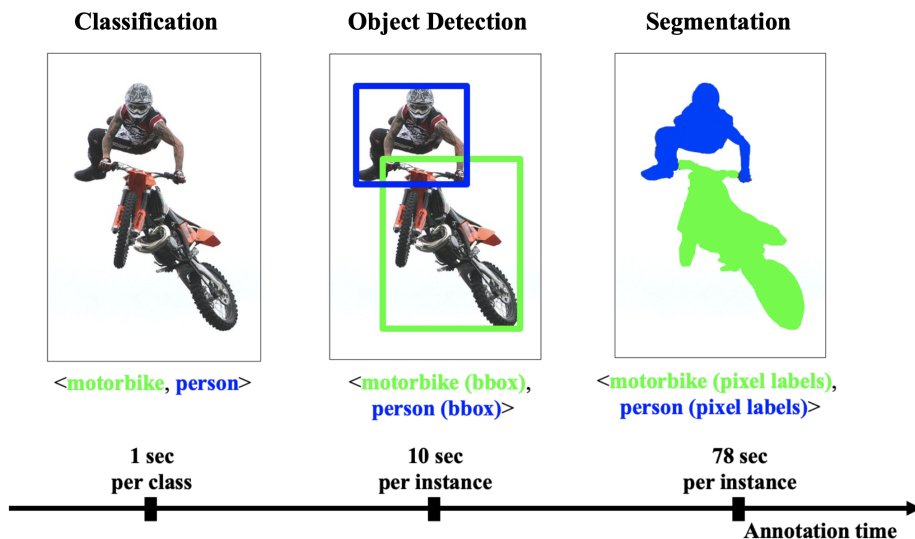


Figure 1.1: Annotation types and times for three different tasks.

Large datasets with complete annotations are essential to the success of the image recognition task [34, 53, 63, 33, 36]. In Fig. 1.1, it shows each annotation type and time for three types of recognition tasks. Among them, labeling for object detection requires a pair of a category and a bounding box (bbox) location for each object within each image, and it is known that it takes about 10 seconds for labeling an object [64, 2]. As such, labeling for object detection consumes enormous costs, time, and effort. As an example, the Caltech pedestrian detection benchmark took about 400 hours to annotate 250k images [21].

Despite the data labeling cost for the object detection tasks being substantially more than that of the classification tasks, semi-supervised learning methods for classification have been mainly studied. Therefore, it is necessary to investigate semi-supervised learning for object detection or segmentation which takes more time due to the instance-level annotation.

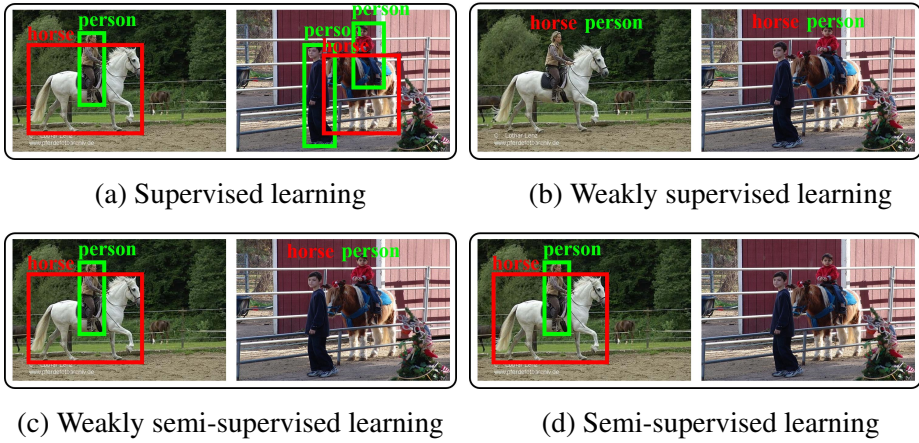


Figure 1.2: Different types of object detection settings

Because of the higher complexity in time and resource for creating object detection datasets, methods for learning with weakly labeled data (D_W) or unlabeled data (D_U) have been recently studied as opposed to learning with the labeled data (D_L)¹ only. There are mainly three types of object detection methods for reducing the cost of such labeling: Weakly-Supervised, Weakly-Semi-Supervised, and Semi-Supervised Learning. Weakly-Supervised Learning trains a model with a dataset that has only class information but no location information (D_W) [96, 65, 29, 80, 30], as shown in Fig. 1.2.(b). Although this takes less effort than the existing box-level labeling method, it results in a far inferior localization performance compared to fully supervised learning [61, 12]. On the other hand, Weakly-Semi-Supervised Learning is a learning method that uses D_W as well as D_L [70, 87], as shown in Fig. 1.2.(c). Weakly-Semi-

¹ $D_L = (I_i, y_i)_{i=1}^{N_L}$ where $y_i = (class^j, bbox^j)_{j=1}^{M_i}$, $D_W = (I_i, y_i)_{i=1}^{N_W}$ where $y_i = (class^j)_{j=1}^{M_i}$, and $D_U = (I_i)_{i=1}^{N_U}$. Here, N_X is the number of images, and M_i is the number of objects in the image I_i .

Supervised detector improves its performance compared to that of Weakly-Supervised Learning, but it still needs to label classes for D_W . In the setting of Semi-Supervised object detection, instead of D_W , unlabeled data D_U is utilized in combination with the labeled data (D_L) [82, 54], as shown in Fig. 1.2.(d). Studies on complete Semi-Supervised object detection have recently been studied [82, 54] which is also the main topic of this dissertation.

The purpose of this dissertation is to propose novel Semi-Supervised Learning for object detection algorithms. We adopt various deep learning-based Semi-Supervised Learning algorithms to object detection. Especially, we considered and solved the problems caused by applying each conventional algorithm directly to object detection. Moreover, we propose a method to combine the proposed algorithms efficiently to improve performance. First, Consistency-based Semi-Supervised Learning for object detection is proposed as a way to improve the detection performance by applying consistency constraint. We define the classification consistency constraint as well as the localization consistency constraint for object detection. Furthermore, Background Elimination that reduces the background effects is suggested.

Second, Interpolation-based Semi-Supervised Learning for object detection, which is an approach by adopting interpolation regularization to improve detection performance, is proposed. We discovered that a problem occurred when IR was directly applied to object detection, and we divided the types according to the case where the problem occurs. Afterwards, we defined the appropriate losses for each type.

Third, efficient way of combining the above two algorithms, CSD and ISD, is proposed. Each CSD or ISD algorithm requires much memory, and simply combining the two algorithms accumulates the memory. In other words, utiliz-

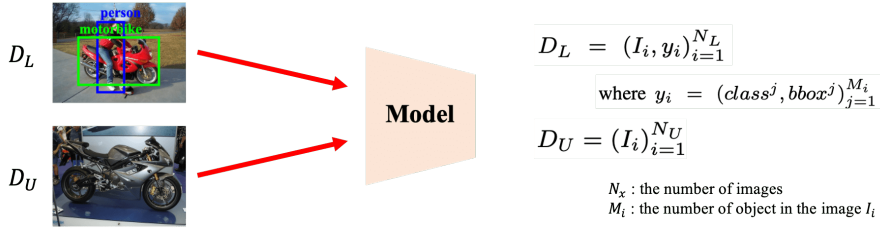


Figure 1.3: Semi-supervised Learning for Object Detection

in the dataset of CSD, the combined model is designed to use the same memory as the ISD.

The remainder of this chapter is organized as follows. In Section 1.1, we define the problem to solve throughout this dissertation. Then, motivation and challenges are discussed in Section 1.2 and Section 1.3, respectively. Contributions of the proposed methods in this dissertation are discussed in Section 1.4. Finally, an outline of the dissertation is given in Section 1.5.

1.1 Problem Definition

This dissertation aims to propose various algorithms that improve the performance of 2D object detection by applying Semi-Supervised Learning schemes. In this dissertation, 2D object detection is defined as the combination of localizing bounding boxes of instances and classifying these regions into one of the pre-defined categories. As shown in Fig 1.3, labeled dataset is composed of $D_L = (I_i, y_i)_{i=1}^{N_L}$ and each annotation of image is consisted of $y_i = (class^j, bbox^j)_{j=1}^{M_i}$ while unlabeled dataset is composed of $D_U = (I_i)_{i=1}^{N_U}$. Here, N_L is the number of images and M_i is the number of objects in the image I_i . The object detector is trained in various ways by using not only D_L but also D_U and the performance of the object detector is verified through the test dataset.

1.2 Motivation

In this section, we discuss the importance of 2D object detection, Semi-Supervised Learning, Semi-Supervised Learning for object detection and the proposed algorithms.

As previously stated, *object detection* predicts where and what objects are in an image. By predicting high-level information from an image, we can apply this algorithm in various fields. For example, through license plate recognition, information on automobiles can be recognized much faster [1, 39, 45]. Also, in the case of a driving aid, an object detector can notify the existence of an object in the driver's blind spot, enabling safer driving. Besides mentioned above, object detection is useful in a wide variety of fields and with a fast and accurate detector, it can change our environment in even better.

Semi-Supervised Learning is a learning method that improves the model's performance by using both labeled and unlabeled data. Since annotation consumes enormous cost, time, and effort, as shown in Fig 1.1, various Semi-Supervised Learning methods have been proposed to reduce the consumption of annotations [77, 84]. Moreover, it shows significant performance improvement compared to conventional supervised learning in the classification task.

Recently, *Semi-Supervised Learning for object detection* research that applies Semi-Supervised Learning to object detection has been introduced. As mentioned before, since the annotation time of object detection is more required than that of classification, the research of Semi-Supervised Learning for object detection is necessary. Nonetheless, Semi-Supervised Learning methods before the pre-deep learning era have been applied to deep learning-based object detection researches.

The *proposed algorithms* are the methods that apply the latest deep learning-based Semi-Supervised Learning method to object detection. By using the latest high-performance Consistency Regularization and Interpolation Regularization mentioned above, a deep learning-based Semi-Supervised Learning method can be applied to object detection leading to efficiency improvement.

1.3 Challenges

There are several obstacles that directly apply classification-based Semi-Supervised Learning methods to object detection. In this section, we discuss the challenges we have faced.

1.3.1 Structure

Object detection predicts multiple bounding boxes and their class probabilities. Recent Semi-Supervised Learning methods for classification tasks adopt multiple inferences, which computes the defined loss from multiple outputs. In classification, It is easy to match each other, but one-to-one correspondence between outputs in object detection is very challenging.

1.3.2 Localization

Object detection predicts not only classification but also localization. As mentioned above, outputs of object detection have a lot of class probabilities and box information which makes it difficult to match each box without annotation. In addition, conventional classification-based Semi-Supervised Learning methods may work well in regression. Moreover, it may be necessary to define a new type of loss for localization.

1.3.3 Background

Unlike classification problems, object detection has an additional class called background. This background causes two problems: The first is the imbalance problem of the number of classes. There are at most 100 instances in an image, but prediction outputs of object detection are more than 30k in [50, 48]. Therefore, semi-supervised learning losses computed with all candidates will be easily dominated by backgrounds. In the absence of ground truth, it is difficult to distinguish the background and apply the algorithm. Second, it causes a problem in interpolation regularization. We found that when interpolation regularization was applied to object detection, the trend was different from that of the conventional classification. When an object is interpolated with a background, the interpolated image appears to be a 100% object corrupted by noise. Therefore, we cannot directly apply interpolation regularization to object detection.

1.3.4 Memory

There is a memory limitation in training object detection. The latest object detectors require large GPU memory to train the model, mainly due to the large batch and network sizes. So memory should be considered even in small model training. However, as mentioned above, the model has to be inferred multiple times, which requires even more memory. Therefore, with our resources (four 1080Ti GPUs), it is difficult to train the state-of-the-art detector algorithms under the same settings, such as the batch and model sizes.

1.4 Contributions

The major contribution of this dissertation is that we are the first to adopt deep learning-based semi-supervised classification algorithms for object detection. Consistency Regularization (CR) and Interpolation Regularization (IR), which are widely used in semi-supervised classification problems, are applied to object detection. Moreover, we identify the problems that appear when CR and IR are applied simultaneously to object detection and solve them, leading to higher performance. Contributions to each method are discussed in this section.

1.4.1 Consistency-based Semi-Supervised Learning for object Detection (CSD)

We propose Consistency-based Semi-Supervised Learning for object Detection (CSD) which is inspired by the Consistency Regularization (CR) [37, 71, 52] that helps train a model to be robust to given perturbed inputs. However, as mentioned in 1.3.1, it is difficult to apply CR directly to the object detection problem in case of multiple candidate boxes are generated for each image. Therefore, we explore the consistency between the box predictions in the original and the horizontally flipped version, which can be simply identified. Then, to tackle the challenge in 1.3.2, we propose a new consistency loss for the location of the predicted boxes, and it shows performance improvement and can help with regression problems. We also propose the Background Elimination (BE) method which excludes boxes with high background probability in the computation of the consistency loss to prevent the ‘background’ class from dominating the consistency loss, which is mentioned in 1.3.3.

We apply our CSD to both the single-stage detector and two-stage detec-

tor and perform various ablation studies to show the benefits of the proposed consistency losses for classification and localization. Also, the effect of BE has been experimentally verified. Our experimental results show that the proposed CSD helps all the detectors improve the performance.

1.4.2 Interpolation-based Semi-Supervised Learning for object Detection (ISD)

We propose Interpolation-based Semi-Supervised Learning for object Detection (ISD), which is inspired by the Interpolation Regularization (IR) [89, 75, 76] that shows the outstanding performance in supervised learning as well as in Semi-Supervised Learning. However, as mentioned in 1.3.3, IR shows different tendency from that of the conventional classification problem. To tackle this problem, we categorize the mixed images into two types depending on whether one of the original images is the object or background. Then, we apply a different Semi-Supervised Learning algorithm suitable for each type.

Our experiments show the effectiveness of the proposed method for each type by demonstrating a significant performance improvement over the conventional algorithms.

1.4.3 Combination of CSD with ISD

We propose the method of combining two algorithms above. In order to combine CSD with ISD, consideration of the memory size is essential. For CSD algorithm, we compute an additional prediction for horizontally flipped images. On the other hand, for the ISD algorithm, we compute two additional predictions for other images and mixed images. Therefore, it requires four times memory size to combine CSD with ISD. Our total memory in GPU available in the lab

is 44 GB (four 1080Ti GPUs), and 12GB of memory is required for conventional SSD. To sum up, when we directly combine two algorithms, it requires 48 GB memory which apparently exceeds our available memory. To solve this problem, we shuffled horizontal flipped images in CSD and mixed with original image batch. With this method, we can get four outputs (original outputs, flipped outputs, shuffled flipped outputs, mixed outputs) using shuffle with three inference (original images, flipped images, mixed images), which enables us to train together in our environment.

In conclusion, combining of CSD with ISD shows much higher performance than each algorithms adopted individually.

1.5 Outline

The structure of this dissertation is composed as follows: In Chapter 2, prior works related to Semi-Supervised Learning and object detection are reviewed. The proposed algorithms for Semi-Supervised Learning for object detection are discussed through Chapter 3 to Chapter 5. Chapter 3 proposes Consistency-based Semi-Supervised Learning for object Detection. Chapter 4 presents Interpolation-based Semi-Supervised Learning for object Detection. Chapter 5 describes the combination method for CSD and ISD. Finally, Chapter 6 provides concluding remarks, limitations, and future directions of this research.

Chapter 2

Related works

This chapter provides related works on Semi-Supervised Learning and object detection.

In Section 2.1, Semi-Supervised Learning is described. In more detail, Section 2.1.1 introduces self-training that is long used for Semi-Supervised Learning. Then, Section 2.1.2 presents Consistency Regularization, which is related to Chapter 3. Section 2.3 represents Interpolation Regularization, which is related to Chapter 4.

In Section 2.2, object detection is described. In more detail, we explain the object detection dataset in Section 2.2.1. Then, evaluation metrics for object detection are presented in Section 2.2.2. Section 2.2.3 represents various object detection algorithms with Supervised and Semi-Supervised Learning. In Section 2.2.3, we further introduce our previous work on supervised object detection.

2.1 Semi-supervised learning

In a real environment, a finite number of labeled data ($\mathcal{L} = \{(x_l, y_l)\}$) is usually provided with an unlimited number of unlabeled data ($\mathcal{U} = \{x_u\}$). Due to this insufficient annotations of real-world sample, a lot of researchers have tried to exploit the potential of unlabeled data.

2.1.1 Self-Training

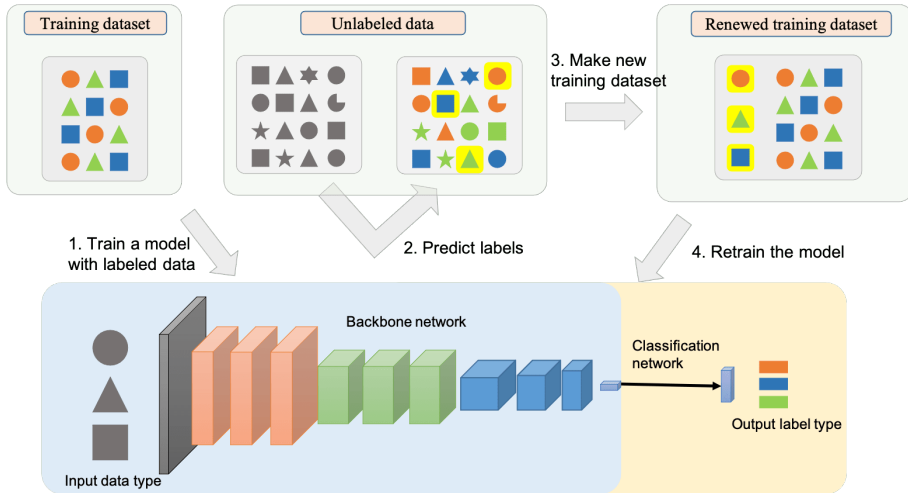


Figure 2.1: The overall structure of Self-Training.

The self-training method has long been researched for Semi-Supervised Learning [51, 62, 93, 95, 86]. It is a resampling technique that repeatedly annotates unlabeled training samples based on the predicted confidence scores and retrains itself with the selected data. Fig. 2.1 shows the overall process of self-training. Self-training methods train a model using labeled data and then make predictions on unlabeled data. If the top-1 prediction score for the input x_u is

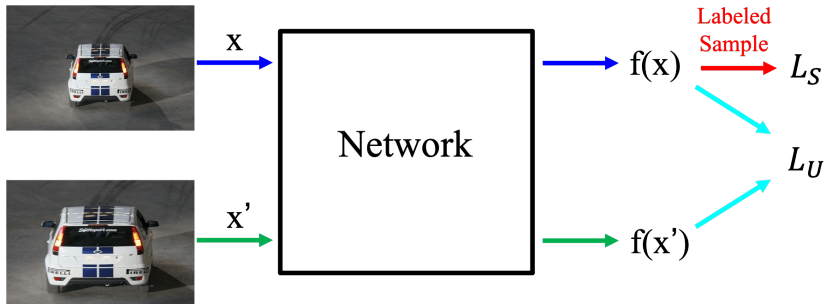


Figure 2.2: The overall structure of Consistency Regularization.

greater than a threshold σ , the pseudo label of x_u is set as the class \bar{y} whose score is the maximum. Then x_u can be treated as a labeled data in the form of (x_u, \bar{y}) [9]. Repetitively applying this process can boost the model’s performance but impedes the whole training speed. In addition, depending on the threshold value σ , the amount of added data varies a lot, resulting in unstable performance. A small number of additional pseudo-labeled samples may not improve the performance sufficiently, while too many samples may degrade the performance with incorrect labeling.

2.1.2 Consistency Regularization

The central idea of the Consistency Regularization methods is to enforce that the model predictions should be the same under reasonable perturbations to the input, as shown in Fig. 2.2 [37, 52, 71]. For object classification, such perturbations involve random translation, random cropping, and horizontal flipping, etc. Let us assume that x_u and x'_u are the original and the perturbed inputs, $d(\cdot, \cdot)$ be a distance function, $w(t)$ be a weighting function over iterations t and $f(\cdot)$ be a function on which consistency loss is measured, then the consistency loss L_U

is computed in an unsupervised manner and consequently the total loss L_{total} is given by a linear combination of the consistency loss and the supervised loss L_S as follows:

$$L_U = d(f(x_u), f(x'_u)) \quad (2.1)$$

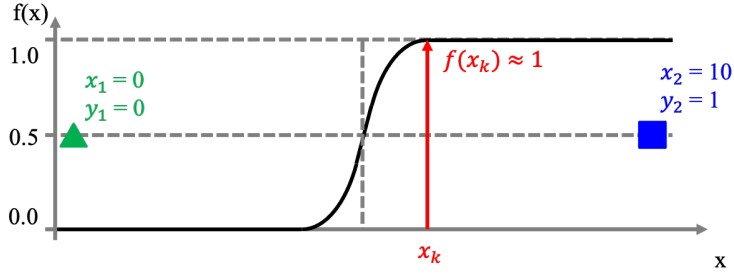
$$L_{total} = L_S + w(t) \cdot L_U. \quad (2.2)$$

Some notable examples of consistency training include the Π model [37] and Mean Teacher [71]. In the case of the Π model, it is a method of learning using different images x_u and x'_u in the same model. In the case of the Mean Teacher, it is a method of learning using the teacher (f_t) and student (f_s) models. At this time, f_t is updated with EMA (exponential moving average).

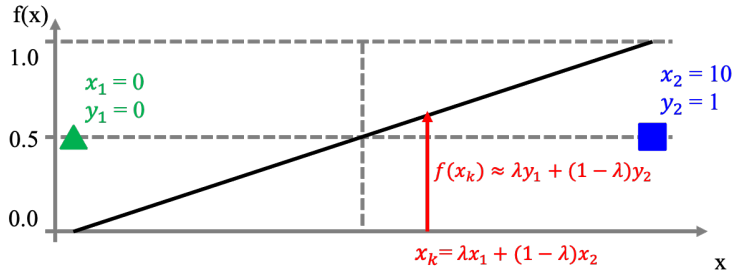
FixMatch [67] is an algorithm that combines self-training and consistency regularization. FixMatch utilizes Weak Augmentation data with horizontal flipping, random translation, and cropping. When the output of Weak Augmentation data exceeds the threshold, it makes a pseudo-label in the same way as in 2.1.1. Then, for the pseudo-labeling samples, it applies strong augmentation [16, 13, 4] and trains with pseudo-labeling. It has state-of-the-art performance among the latest algorithms and is the inspiration for our ISD model.

2.1.3 Interpolation Regularization

An Interpolation-based Regularization is a promising approach due to its state-of-the-art performances and virtually no additional computational cost [89, 75, 31]. These methods construct additional training samples by combining two or more training samples. Mixup [89] and Between-class learning [73] are the earliest works that took steps in this direction. These methods are based on the principle that the output of a supervised network for an affine combination of



(a) Conventional Training



(b) Mixup Training

Figure 2.3: Example of Conventional training and Mixup training methods in binary classification problem

two training samples should change linearly.

Such kind of inductive bias can be induced in a network by training it on the synthetic samples constructed by mixing two samples and their corresponding targets.

$$\begin{aligned} \tilde{x} &= \lambda \cdot x_i + (1 - \lambda) \cdot x_j \\ \tilde{y} &= \lambda \cdot y_i + (1 - \lambda) \cdot y_j \end{aligned} \tag{2.3}$$

As shown in Fig. 2.3.(a), the decision boundary becomes steep under the conventional training scheme. Therefore, for the new data input x_k (red point in Fig. 2.3.(a)) between x_1 and x_2 , the model predicts to 1 even though x_k is







| | | Cutout | Cowout | Label |
|-------|---|---|---|---------------------------------------|
| Image |  |  |  | Bird : 1.0 |
| | Mixup | Cutmix | Cowmix | |
| Image |  |  |  | Bird : λ Dog : $1-\lambda$ |

Figure 2.4: Overview of the images of Cutout, Cowout, Mixup, CutMix, and our CowMix

near the decision boundary. On the other hand, as shown in Fig. 2.3.(b), using the Mixup training method, the model predicts more linearly, and it makes the decision boundary more smooth.

Manifold Mixup [75] mixes features in the deeper layers instead of input images. As shown in the Fig. 2.4, other works such as CutMix [88] and CowMix [24] construct the synthetic samples by mixing the CutOut [16] and CowOut versions of two samples. Overall, these approaches can be interpreted as a form of data-augmentation technique that seeks to construct additional training samples by combining two or more samples.

In the Semi-Supervised Learning setting, Interpolation Consistency Training (ICT) is the approach that applies Interpolation regularization [76]. As shown in Fig. 2.5, ICT encourages the prediction ($f(\text{Mix}_\lambda(x_{u_i}, x_{u_j}))$)¹ at an interpolation of unlabeled samples ($(\text{Mix}_\lambda(x_{u_i}, x_{u_j}))$) to be consistent with the inter-

¹ $\text{Mix}_\lambda(A, B) = \lambda \cdot A + (1 - \lambda) \cdot B$

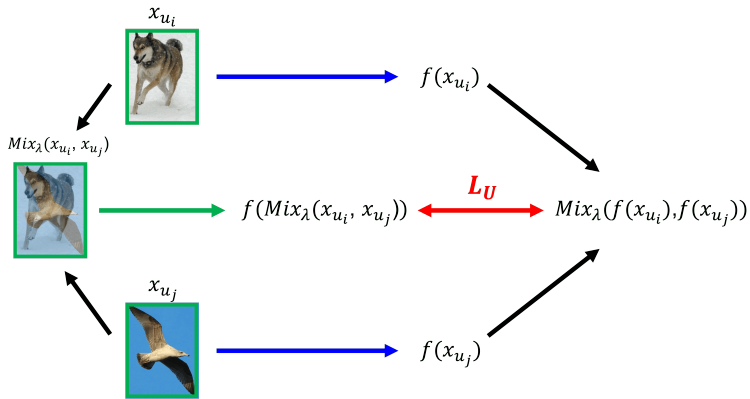


Figure 2.5: Interpolation Consistency Training

pulation ($Mix_{\lambda}(f(x_{u_i}), f(x_{u_j}))$) of the predictions at those samples ($f(x_{u_i}), f(x_{u_j})$). It achieves state-of-the-art performance, and we adopt this approach to object detection in chapter 4.

2.2 Object detection

2.2.1 Dataset

Table 2.1: Pascal VOC and MSCOCO Dataset

| | PASCAL VOC | MSCOCO |
|---|---|---|
| # of classes | 20 | 80 |
| Super Category (# of classes in Super Category) | Person (1) Animal (6) Indoor (6) Vehicle (7) | Person (1), Accessory (5) Animal (10), Appliance (5) Electronic (6), Food (10) Furniture (6), Indoor (7) Kitchen (7), Outdoor (5) Sports (10), Vehicle (8) |
| Train dataset | VOC2007 trainval VOC2012 trainval | COCO Train2014 COCO Val2014-35k |
| Test dataset | VOC2007 test | COCO test-dev |

In this dissertation, we have utilized the PASCAL VOC [23] and MSCOCO [49] datasets which are the most popular datasets in object detection. They consist of 20 and 80 classes, respectively. PASCAL VOC 2007 and 2012 datasets consist of 5k and 12k trainval (train and validation) images respectively. COCO Train2014 and COCO Val2014-35k datasets consist of 83k and 35k images, respectively. We use a test set of PASCAL VOC2007 (5k images) and MS COCO test-dev (20k images) for testing.

Table 2.2: Pascal VOC and MSCOCO Classes (**Bold** : the intersection classes between two datasets)

| Super Category | PASCAL VOC | MSCOCO |
|----------------|--|---|
| Person | Person | Person |
| Accessory | | Backpack, Handbag, Suitcase, Tie, Umbrella |
| Animal | Bird, Cat, Cow, Dog, Horse, Sheep | Bear, Bird, Cat, Cow, Dog , Elephant, Giraffe, Horse, Sheep , Zebra |
| Appliance | | Microwave, Oven, Refrigerator, Sink, Toaster |
| Electronic | | Cell phone, Keyboard, Laptop, Mouse, Remote, TV |
| Food | | Apple, Banana, Broccoli, Cake, Carrot, Donut, Hot dog, Orange, Pizza, Sandwich |
| Furniture | | Bed, Chair, Couch , Dining table, Potted plant , Toilet |
| Indoor | Bottle, Chair, Dining table, Potted plant, Sofa, Tv/monitor | Book, Clock, Hair drier, Scissors, Teddy bear, Toothbrush, Vase |
| Kitchen | | Bottle , Bowl, Cup, Fork, Knife, Spoon, Wine glass |
| Outdoor | | Bench, Fire hydrant, Parking meter, Stop sign, Traffic light |
| Sports | | Baseball bat, Baseball glove, Frisbee, Kite, Skateboard, Skis, Snowboard, Sports ball, Surfboard, Tennis racket |
| Vehicle | Aeroplane, Bicycle, Boat, Bus, Car, Motorbike, Train | Airplane, Bicycle, Boat, Bus, Car, Train , Truck, Motorcycle |

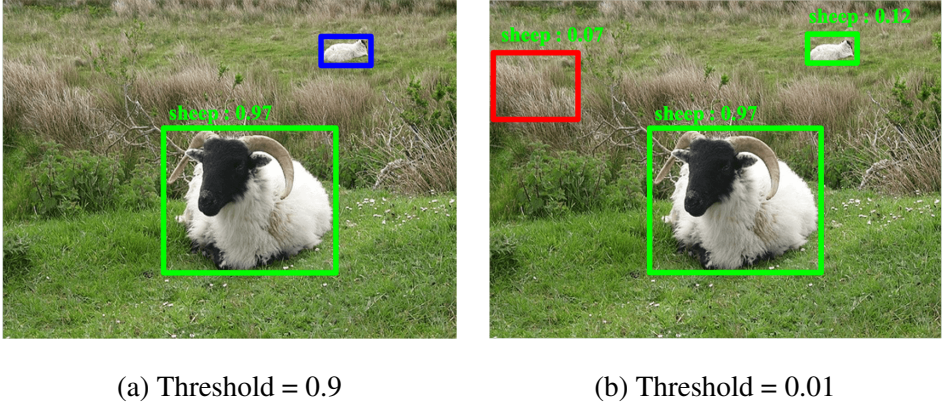


Figure 2.6: Detection results according to threshold. **Blue box** represents the false negative and **Red box** represents the false positive.

2.2.2 Evaluation metric

We use mean Average Precision (mAP) as an evaluation metric. Since object detection performance is a matter of detecting objects rather than classification, the Average Precision (AP) evaluation metric is widely used. AP is a concept of integrating precision as recall is varied from 0 to 1. And mAP is defined as the average of AP for all the object classes.

As shown in Fig. 2.6.(a), the detection accuracy may be high with a high threshold, but there is a false negative (Blue Box) that is not recognized by the detector. In this case, it has high precision and low recall. On the other hand, as shown in Fig. 2.6.(b), with a low threshold, all objects are detected, but there is a false positive (Red Box) that is detected as an object, but it is not. In this case, it has a high recall but low precision. The precision and recall are computed as follows:

$$Recall = \frac{TP}{TP + FN}, \quad Precision = \frac{TP}{TP + FP} \quad (2.4)$$

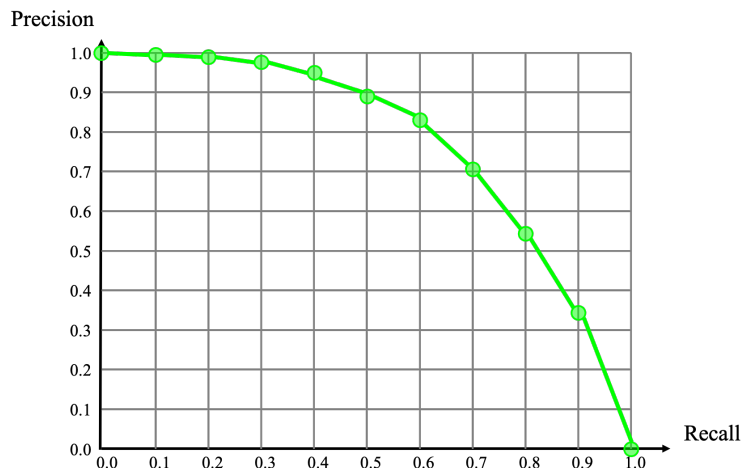


Figure 2.7: Recall vs. Precision graph

Here, TP is a true positive, FN is a false negative, and FP is a false positive.

In Fig 2.7, we show the example of recall vs. precision graph. Most of the algorithms show decreasing the precision according to increase the recall. And, a detector with high precision in many candidates (high recall) will have high AP. In other words, higher mAP means better detection accuracy.

The criterion for true positive of the prediction box is Intersection over Union (IoU) overlap between ground truth and prediction bounding boxes. In the Pascal VOC dataset, the prediction box is regarded as a true positive under the criterion of $\text{IoU} \geq 0.5$. In the MSCOCO dataset, evaluation is conducted under various IoU criteria. $\text{AP}@0.5$ is the same as the criterion of Pascal VOC, and $\text{AP}@0.75$ has a stricter standard with $\text{IoU} \geq 0.75$. $\text{AP}@0.5:0.95$ is the average AP for IoU from 0.5 to 0.95 with a step size of 0.05

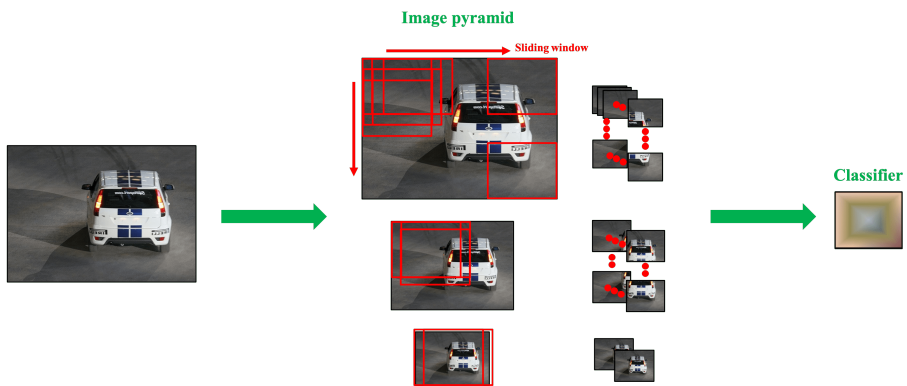
2.2.3 Survey of Object Detection Algorithms

Object detection has been researching with a wide variety of approaches. Until a recent date, these methods can be broadly categorized into (1) Single-Stage methods and (2) Two-Stage methods.

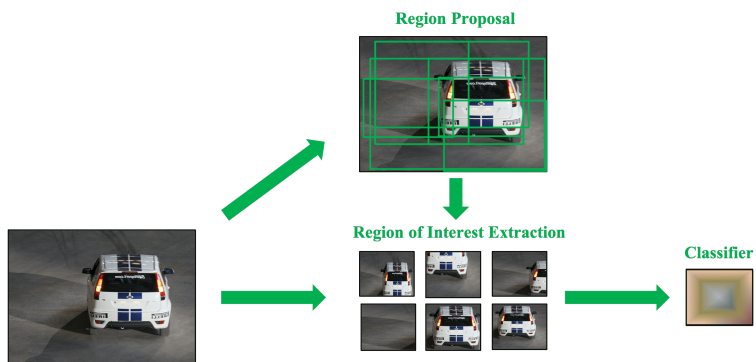
Fig. 2.8 shows the single-stage and two-stage methods in the pre-deep learning era. The single-stage method has a fixed size classifier, as shown in Fig. 2.8.(a), which slides the image and classifies all positions [15, 19, 20, 38]. In this case, since it is impossible to detect multiple scales, the image pyramids are generated. The classifier slides the resized images and classifies them for all scales. As shown in Fig. 2.8.(b), the two-stage method extracts the location where the object is likely to be using segment information² and classifies for the extracted samples [74, 97].

In comparing the two methods, the single-stage method enables faster classification, and the two-stage method provides more accurate predictions. Although the single-stage method has to perform classification for all candidates, it does not take a long time [18, 19, 17]. The classifier does not require much computation, and it is possible to classify using some algorithms such as soft cascade quickly [91]. Among the single-stage methods, an algorithm capable of 100 Frame Per Second (FPS) has been introduced [3], which means that 100 images can be detected in 1 second. On the other hand, in the two-stage method, the number of candidate samples is much smaller, and the classification speed is the same. However, it took tremendous time to calculate these regions of interest. As an example above, there is a Selective search algorithm [74], which has Fast and Quality versions. In the case of the Fast version, it takes about 3.7

²Groups are created with similar values of surrounding pixels, and boxes are generated using the grouped edge information.



(a) Single Stage Method



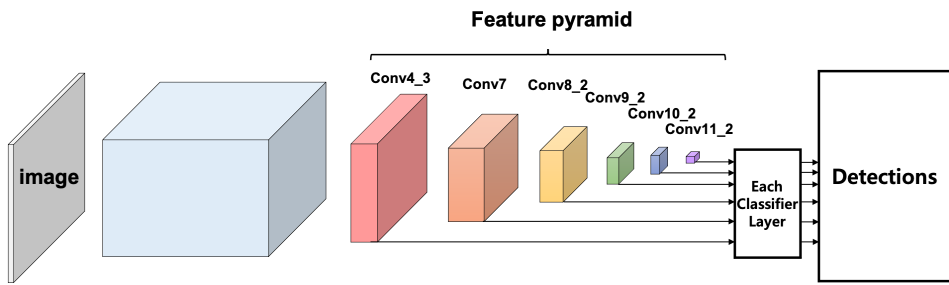
(b) Two Stage Method

Figure 2.8: Two types of object detectors in pre-deep learning era.

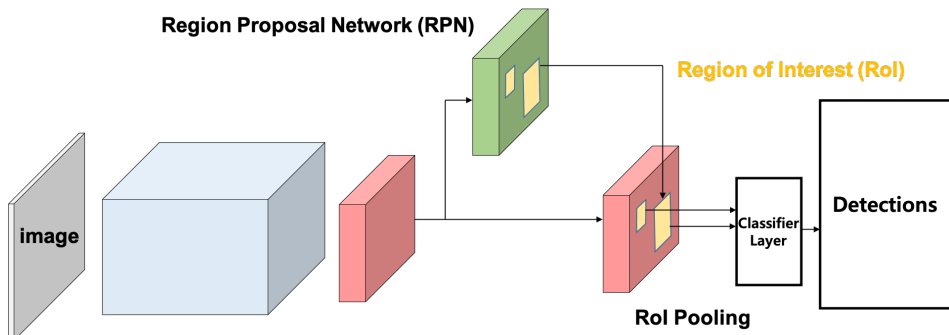
seconds to sample about 2k box candidates, and in the case of the Quality version, it takes about 17 seconds to sample about 10k candidates. Therefore, it is not easy to detect in real-time in the two-stage method. In terms of detection accuracy, in the single-stage method, accurate regression is difficult due to the size of the specified classifier. On the other hand, in the two-stage method, accurate regression is possible because the detection is performed according to the candidate group for the region of interest.

Even deep learning was applied to object detection, the above two schemes were similarly applied. The single-stage detector has changed from the image pyramid to the feature pyramid [50] or grid method [57], as shown in Fig. 2.9(a). And, the two-stage detector has changed from Selective Search to Region Proposal Network (RPN) [60], as shown in Fig. 2.9(b). Single-stage detectors perform classification and localization in all the spatial locations of feature maps. On the other hand, Two-stage detectors are RPN-based algorithms, which detect objects only for RoIs that have a high possibility of containing an object [14, 60]. There have been tremendous performance improvements using deep learning, and there are algorithms that are able to detect objects in real-time on a desktop.

In comparing the deep learning-based two detectors, same as predecessors, the single-stage detectors still enable faster classification, and the two-stage detectors provide accurate predictions. The two-stage detector requires an additional inference of RPN and sorting for the sampling. In addition, in the process of classification, convolution or fully connected operation for RoIs consume much redundant operation, which consumes a lot of time. Therefore, many studies to improve the speed of the two-stage detector have been introduced [43, 66]. On the contrary, in the single-stage detector, researchers have focused on per-



(a) Single Stage Detector



(b) Two Stage Detector

Figure 2.9: deep learning based two types of object detectors

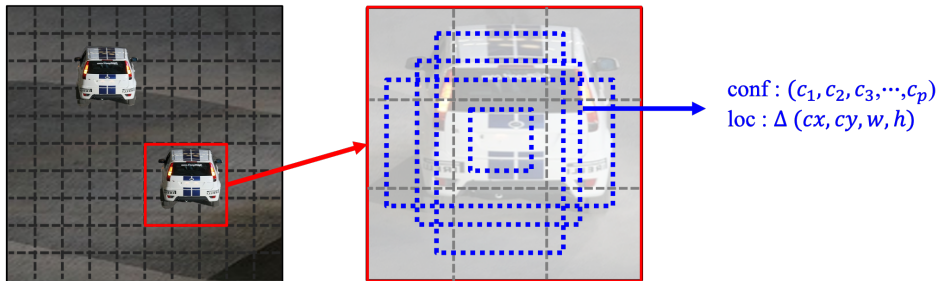


Figure 2.10: Example of classifier of object detection based on deep learning

formance improvement.

In deep learning, the Classifier layer predicts not only classification but also regression for the refinement. Fig. 2.10 shows an example of a classification process in SSD. The left of the figure shows the feature map of 10×10 , and the classifier network operates for the fixed area as shown in the red box. As shown in the middle of the figure, each classifier has several default boxes, and each default box consists of the softmax output vector (c_1, c_2, \dots, c_p) and the localization offset of the center and the size of the box $[\Delta cx, \Delta cy, \Delta w, \Delta h]$. The localization offset is trained in relative coordinates, not absolute coordinates, to take advantage of the shift-invariant property. In other words, the localization is not the position from the upper left coordinate $(0,0)$, but the difference of center position and size of the object from the default box. In the two-stage detector, an anchor box is applied to extract RoIs in RPN.

2.2.3.1 Supervised Learning

A wide variety of methods using deep learning have been applied to the problem of object detection and it continues to show performance improvements [57]. In the earlier works pioneered by R-CNN (region-based CNN) [27], the candidate

region was proposed through a separate algorithms such as selective search [74] or Edge boxes [97] and the classification was performed with deep learning.

Although R-CNN improved the accuracy using deep learning, speed was still a problem, and end-to-end learning was impossible. The region proposal network (RPN) was first proposed in faster R-CNN, which improved the speed of the object detector significantly and was able to learn end-to-end [60].

YOLO (you only look once) greatly improved the speed by dividing a single image into multiple grids and simultaneously performing localization and classification in each grid [57]. While YOLO performed object detection by concentrating only on speed, an enhanced version of YOLO, which is denoted as YOLO2, removed the fully connected layers and used anchor boxes to improve both the speed and the accuracy [58].

On the other hand, SSD creates bounding box candidates at a given position and scale and obtains their actual bounding box and score for each class [50]. To improve the accuracy of SSD, especially for small object, DSSD (deconvolutional SSD) that uses a large scale context for the feature pyramid was proposed [25]. DSSD applied a deconvolution module to the feature pyramid and used ResNet instead of VGGNet. DSSD succeeded in raising accuracy at the expense of speed. Like DSSD, methods to use the feature pyramid efficiently have been studied [47, 85, 44], and one of them will be introduced in 2.2.3. These are methods of improving model performance by making correlations between feature pyramids.

These object detection algorithms are continuously being studied, such as research on new approaches based on key points [40, 22, 41], research on lightweight the model [59, 11, 10], new methods for high performance [46, 90, 69] and high speed [83, 6], etc.

2.2.3.2 Our previous work in supervised object detection

Enhancement of SSD by concatenating feature maps for object detection

We proposed an object detection method that improves the conventional Single Shot Multibox Detector (SSD) performance, one of the top object detection algorithms in both aspects of accuracy and speed.

Although the conventional SSD performs well in both speed and detection accuracy, it has a couple of points to be supplemented.

First, each layer in the feature pyramid is used independently as an input to the classifier network, as shown in Fig. 2.9.(a). Thus, the same object can be detected on multiple scales. Consider a certain position of a feature map in a lower layer is activated. This information can affect entire scales up to the last layer, which means that the relevant positions in the higher layers have a good chance to be also activated. However, SSD does not consider the relationships between the different scales because it looks at only one layer for each scale. For example, in Fig.2.11(a), SSD finds various scale boxes for one object.

Second, SSD has the limitation that small objects are not detected well. This is not the problem only for SSD but the problem for most object detection algorithms. There have been various attempts to solve this problem such as replacing the base network with more powerful one, e.g., replacing VGGNet with ResNet [14, 25] or increasing the number of channels in a layer [42]. Fig. 2.11(b) shows that SSD has a limitation in detecting small objects. Especially in the two figures, persons on the boat and small cows are not detected, respectively.

In this research, we tackle these problems as follows. First, the classifier network is implemented considering the relationship between layers in the feature pyramid. Second, the number of channels (or feature maps) in a layer is

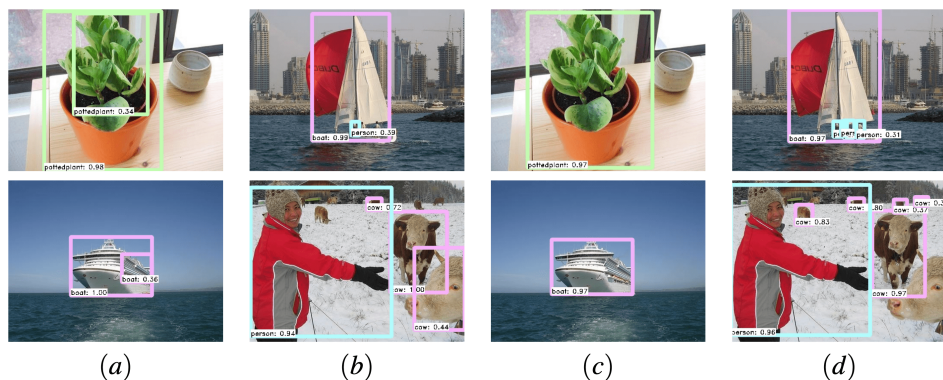


Figure 2.11: Conventional SSD vs. the proposed Rainbow SSD (R-SSD). Boxes with objectness score of 0.3 or higher is drawn: (a) SSD with two boxes for one object; (b) SSD for small objects; (c) R-SSD with one box for one object; (d) R-SSD for small objects

efficiently increased. More specifically, only the layers in the feature pyramid are allowed to have increased number of feature maps instead of increasing the number of layers in the base network. The proposed network is suitable for sharing weights in the classifier networks for different scales, resulting in a single classifier network. This enables faster training speed with advanced generalization performance. Furthermore, this property of a single classifier network is very useful in a small database. In the conventional SSD, if there is no object at a certain size, the classifier network of that size cannot learn anything. However, if a single classifier network is used, it can get information about the object from the training examples in different scales.

Using the proposed architecture, our version of SSD can prevent detecting multiple boxes for one object, as shown in Fig. 2.11(c). In addition, the number of channels can be efficiently increased to detect small objects, as shown in

Fig. 2.11(d). At that time, the proposed method showed state-of-the-art mAP with a slightly degraded speed compared to the conventional SSD.

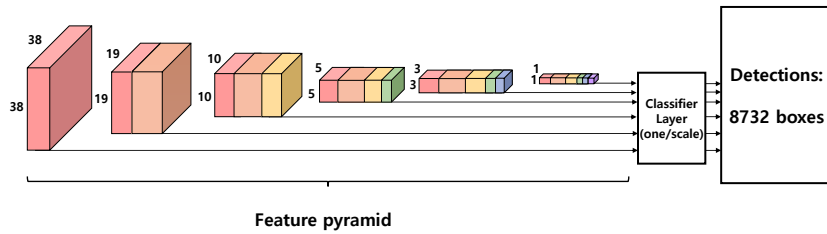
Method

Our strategy of improving the accuracy of SSD is to make the classifier network fully utilize the relationship between the layers in the feature pyramid without changing the base network that is closely located to the input data. In addition, it also increases the number of channels in the feature pyramid efficiently.

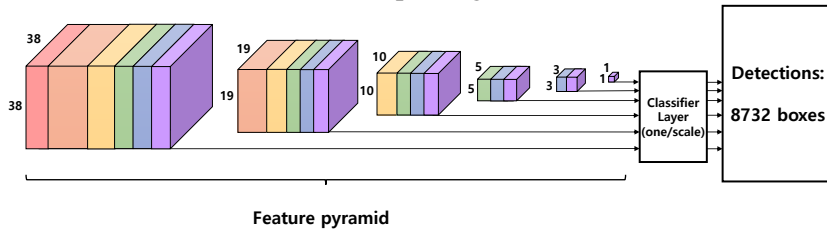
Fig. 2.12 shows several ways of increasing the number of feature maps in different layers for the classifier networks to utilize the relationship between layers in the feature pyramid. To enable this, in Fig. 2.12(a), feature maps in the lower layers are concatenated to those of the upper layers through pooling. In this way, the classifier networks with large receptive fields can have enriched representation power for object detection. On the other hand, Fig. 2.12(b) shows the method of concatenating the feature maps of the upper layers to the lower layer features through deconvolution or upsampling. Fig. 2.12(c) shows the feature map concatenation method that utilizes both the lower layer pooling and the upper layer deconvolution.

One thing to note is that before concatenating feature maps, a normalization step is inevitable. This is because the feature values in different layers are quite different in scale. Here, normalization is applied for each filter before concatenation.

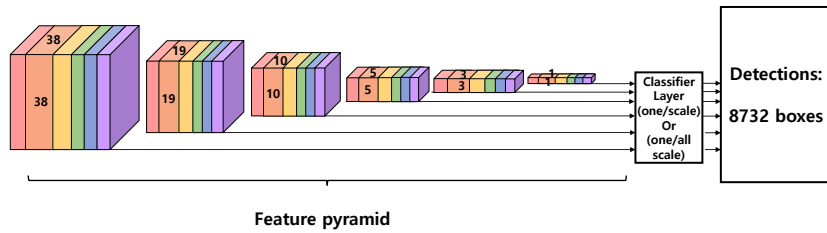
All of the above methods have the advantage that end-to-end learning is possible. More details about each are described below.



(a) pooling



(b) deconvolution



(c) both pooling and deconvolution (Rainbow concatenation)

Figure 2.12: Proposed methods of feature concatenation: (a) concatenation through pooling (b) concatenation through deconvolution; (c) rainbow concatenation through both pooling and concatenation.

1) Concatenation through pooling or deconvolution

In the structure of SSD, generally, the numbers of channels in the lower layers are larger than those in the upper layer. To make an explicit relationship between the feature pyramid and to increase the number of channels effectively, we concatenate feature maps of the upper layers through pooling or concatenate feature maps of the lower layers through deconvolution. Unlike DSSD [25], which uses a deconvolution module consisting of 3 convolution layers, 1 deconvolution layer, 3 batch normalization layers, 2 Relu and elementwise products, our model of concatenation through deconvolution performs the only deconvolution with batch normalization and does not need the elementwise product.

The advantage of these structures is that object detection can be performed with information from the other layers. On the other hand, the disadvantage is that information flows unidirectional, and the classifier network cannot utilize other directional information.

2) Rainbow concatenation

As shown in Fig. 2.12(c), in the rainbow concatenation, pooling and deconvolution are performed simultaneously to create feature maps with an explicit relationship between different layers. After pooling or deconvolving features in every layer to the same size, we concatenate them. Using these concatenated features, detection is performed considering all the cases where the size of the object is smaller or larger than the specific scale. That is, it can have additional information about the object larger than or smaller than the object. Therefore, it is expected that the object in a specific size is likely to be detected only in an appropriate layer in the feature pyramid, as shown in Fig. 2.11(c).

In addition, the low-layer features that have been with limited representation power are enriched by the concatenation of higher-layer features, resulting in good representation power for small object detection as in DSSD [25] without much computational overhead.

Experiment and Discussion

Experiment : We trained our model with VOC2007 and VOC2012 ‘trainval’ datasets. In the case of speed, it is measured by using the forward path of the network with a batch size of 1. The experiments were done with cuDNN v5.1 using CAFFE time function. Therefore, if the detection time is measured from the pre-processing (resizing image and so on), it may take longer. The experimental results are shown in Table 2.3. In the table, the performances of YOLO [57], YOLOv2 [58], Faster R-CNN [60], R-FCN [14], and DSSD [25] were obtained from their homepage ³ or the respective paper. To see the performance of various feature augmentation methods, we performed experiments using features concatenated through pooling (SSD pooling) and deconvolution (SSD deconvolution) as described in Section 2.2.3. Three types of R-SSD was tested. The first one utilizes separate classifier networks for different scales and the rest two use a common classifier with 4 or 6 default boxes for a scale as described in Section 2.2.3. The conventional SSD was also trained and tested by ourselves.

For the 300 input model, there is a 0.8% improvement in accuracy with 78.5% mAP compared to conventional SSD. However, due to the increased computational complexity, the speed drops to 35.0 FPS. For the 512 input model, it results in mAP of 80.8% which is 1% better than conventional SSD. However its speed drops to 16.6 FPS. In particular, comparing the two SSD 512 models,

³YOLO and YOLOv2 : <http://pjreddie.com/darknet>

| | Input | Train | Test | mAP | FPS |
|--|-------|--------------|------|-------------|------|
| YOLO[57] | 448 | VOC2007+2012 | 2007 | 63.4 | 45 |
| YOLOv2[58] | 416 | VOC2007+2012 | 2007 | 76.8 | 67 |
| YOLOv2 544x544[58] | 544 | VOC2007+2012 | 2007 | 78.6 | 40 |
| Faster R-CNN[60] | | VOC2007+2012 | 2007 | 73.2 | 5 |
| R-FCN (ResNet-101)[14] | | VOC2007+2012 | 2007 | 80.5 | 5.9 |
| SSD*[50] | 300 | VOC2007+2012 | 2007 | 77.7 | 61.1 |
| DSSD (ResNet-101)[25] | 321 | VOC2007+2012 | 2007 | 78.6 | 9.5 |
| ISSD* | 300 | VOC2007+2012 | 2007 | 78.1 | 26.9 |
| ours (SSD pooling)* | 300 | VOC2007+2012 | 2007 | 77.1 | 48.3 |
| ours (SSD deconvolution)* | 300 | VOC2007+2012 | 2007 | 77.3 | 39.9 |
| ours (R-SSD)* | 300 | VOC2007+2012 | 2007 | 78.5 | 35.0 |
| ours (R-SSD one classifier (4 boxes))* | 300 | VOC2007+2012 | 2007 | 76.2 | 34.8 |
| ours (R-SSD one classifier (6 boxes))* | 300 | VOC2007+2012 | 2007 | 77.0 | 35.4 |
| SSD*[50] | 512 | VOC2007+2012 | 2007 | 79.8 | 25.2 |
| DSSD (ResNet-101)[25] | 513 | VOC2007+2012 | 2007 | 81.5 | 5.5 |
| ours (R-SSD)* | 512 | VOC2007+2012 | 2007 | 80.8 | 16.6 |

Table 2.3: VOC2007+2012 training and VOC 2007 test result (* is tested by ourselves)

precision increases 2.9% at recall value 0.8 and 8.2% at recall of 0.9. In the case of single classifier model with 300 input model, it has a 76.2% and 77.0% mAP when they use four and six default boxes, respectively.

Concatenation by pooling or deconvolution : These two models are both inferior in accuracy and speed compared to conventional SSD, although they made explicit relationship between multiple layers and increased the number of channels. These two models need to perform more operations, therefore the speed can drop. As for accuracy, the reason can be conjectured that the layers sharing the same feature maps with other layers can be affected by the loss of other scales and do not fully focus on the scale. That is, they cannot learn properly on their scale.

Single classifier vs. Multiple classifiers : Unlike the conventional SSD, because R-SSD have the similar feature maps for different layers only different in size, the classifier network can be shared. Here, the experiments with a single classifier network by unifying the number of channels in each scale of feature pyramid. As shown in the Table 2.3, there is a difference in the number of boxes, but there is little difference in speed. In comparison, performance was 1.2 % and 0.7 % lower than that of conventional SSD. However, the advantage of a single classifier is that learning can be effective especially when there are significant imbalance between the numbers of training samples for different sizes. In this case, conventional SSD cannot train the classifier for a scale with small number of samples. However, in R-SSD, this problem is avoided because the classifier network is shared. Furthermore, single classifier is faster at the early stage of training. Therefore, even for a large dataset, R-SSD can be trained fast by train-

| | Recall (# of detected objects / # of total object) | | |
|----------------|--|--|--------------------------------|
| | Small (area < 32 ²) | Medium (32 ² < area < 96 ²) | Large (96 ² < area) |
| SSD300 | 0.3845 (218/567) | 0.7754 (2965/3824) | 0.9314 (7117/7641) |
| ours(R-SSD300) | 0.4127 (234/567) | 0.8073 (3087/3824) | 0.9374 (7163/7641) |
| SSD512 | 0.6526 (370/567) | 0.8023 (3068/3824) | 0.9361 (7153/7641) |
| ours(R-SSD512) | 0.6949 (394/567) | 0.8248 (3154/3824) | 0.9365 (7156/7641) |

Table 2.4: Recall for objects in different size [49]. A box is declared as an object when the object score is higher than 0.1.

ing a single classifier in the early stage and at some point, the classifiers can be trained separately for different scales.

Accuracy vs. Speed : The conventional SSD is one of the top object detection algorithms in both aspects of accuracy and speed. For SSD300 or SSD512, it has 77.7 % mAP and 79.8 % mAP respectively and has 61.1 FPS and 25.2 FPS. Looking at the results of the ISSD for comparison with our algorithm, ISSD had a 0.4 % accuracy gain, but the speed dropped to 26.9 FPS. In our experiments, R-SSD shows improved accuracy with a bit slow speed. Compared to the ISSD, R-SSD shows higher accuracy and faster speed. Moreover, it shows about 1% mAP improvement over the conventional SSD. At a speed of 15 fps or higher, our R-SSD shows 80.8 % mAP. Compared to R-FCN with similar accuracy, R-SSD is about three times faster.

Performances for different scales : Table 2.4 shows the recall of each object size [49]. Normally, the AP or AR (Average Recall) should be obtained, but, the VOC2007 test set has a total of 12,032 objects, of which 567 are small ob-

| | Input | pre-trained model | mAP |
|---------------------------------------|-------|-------------------|------|
| SSD [50] | 300 | reduced VGG-16 | 66.1 |
| ours (R-SSD) | 300 | reduced VGG-16 | 66.9 |
| ours (R-SSD one classifier (6 boxes)) | 300 | reduced VGG-16 | 67.2 |

Table 2.5: Results on VOC2007 test dataset trained with VOD2007 small train dataset (2,501 images)

jects. In evaluating the performance for small objects, there are several classes with no object at all. Therefore, we integrate all the objects in measuring the recall. When the object size is small, R-SSD300 and R-SSD512 detect more number of objects than SSD300 and SSD512, respectively. It can be shown that R-SSD misses a few small objects. At medium size, R-SSD300 has even more recall than SSD512. Furthermore, when the object size is large, recall of all models show high value over 0.93. The difference of R-SSD300, SSD512, and R-SSD512 is less than 10 out of 7,641.

Small Train Dataset in VOC2007 : Table 2.5 is experimental results of different networks that were trained using the *train dataset* in VOC2007 which consists of a relatively small number of images (2,501 images in total). Each network was trained with these 2,501 images and the mAP was measured with VOC 2007 test dataset. The conventional SSD [50] shows a mAP of 66.1%, while R-SSD achieves 66.9% which is 0.8% better than that of SSD. Furthermore, R-SSD with one classifier achieves an even better mAP of 67.2%. It shows that training a single classifier is better not only in generalization power resulting in a faster training speed but also in detection performance when the training

dataset is small.

Conclusion

In this research, we presented a rainbow concatenation scheme that can efficiently solve the problems of the conventional SSD. The contribution of the paper is as follows. First, it creates a relationship between each scale of feature pyramid to prevent unnecessary detection such as multiple boxes in different scales for one object. Second, by efficiently increasing the number of feature maps of each layer in the feature pyramid, the accuracy is improved without much time overhead. Finally, the number of feature maps for different layers are matched so that a single classifier can be used for different scales. By using a single classifier, improvement on the generalization performance can be expected, and it can be effectively used for datasets with size imbalance or for small datasets. The proposed R-SSD was created considering both the accuracy and the speed simultaneously, and showed state-of-the-art mAP among the ones that have speed of more than 15 FPS.

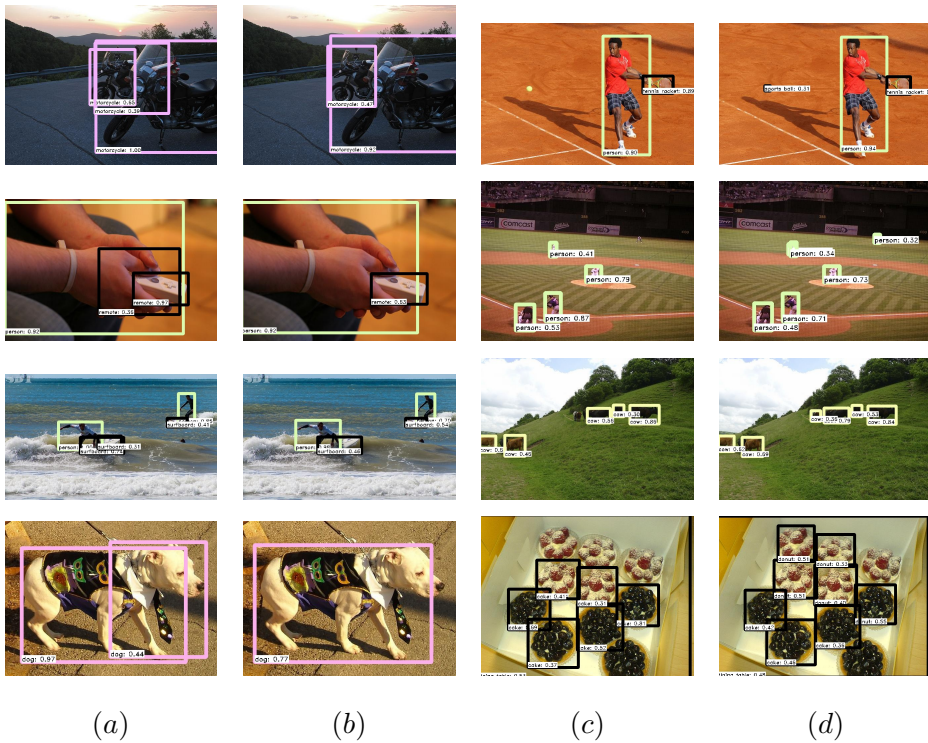


Figure 2.13: Conventional SSD vs. the proposed Rainforest SSD (R-SSD). Boxes with objectness score of 0.3 or higher is drawn: (a) SSD with two boxes for one object; (b) R-SSD with one box for one object (c) SSD for small objects; (d) R-SSD for small objects;

2.2.3.3 Semi-supervised learning

Until recently, most semi-supervised learning methods for object detection are based on the self-training scheme [82, 81]. A representative method is the Self-supervised Sample Mining (SSM) [82] algorithm. It trains a model with labeled data and then predicts unlabeled data. For high confidence patches predicted in unlabeled data, SSM makes pseudo-labeling and stitches those patches to labeled dataset. And, the model is re-trained with new labeled dataset. SSM performs the above method until no new sample is added. SSM has used a method called ‘evaluating consistency’ to make the pseudo box label robust. It operates as a mask to verify that the Intersection over Union (IoU) score between the previously detected box and the currently detected box is greater than threshold γ . Therefore, SSM differs from our method of directly using consistency losses. SSM repeats the process of making an intermediate unlabeled data prediction and changing the training set. Consequently, SSM shares the same drawback as self-training.

Chapter 3

Consistency-based Semi-supervised learning for object Detection (CSD)

3.1 Introduction

In this chapter, we introduce a Consistency-based Semi-supervised learning for object Detection (CSD) which is similar to the consistency regularization (CR) [37, 71, 52] that has shown state-of-the-art performance in semi-supervised classification [55]. CR helps train a model to be robust to given perturbed inputs. However, as shown in Fig. 3.1, it is difficult to apply CR directly to the object detection problem where multiple candidate boxes are generated for each image. Because images with different perturbation may have different numbers of boxes with various locations and sizes, it is difficult to match boxes in given images. Therefore, we use the horizontally flipped image so that one-to-one correspondence between the predicted boxes in the original and the flipped images can be easily identified. In our method, in addition to applying consistency constraint to the classification results for each predicted box, we propose a new consistency loss for fine-tuning the location of the predicted box. Experimental

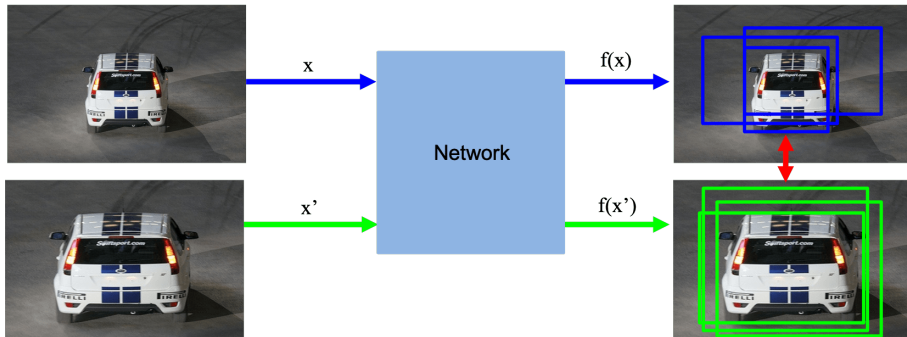


Figure 3.1: difficult to establish a one-to-one correspondence

results show that each of these consistency losses can improve performance and we can get additional performance improvement by combining these two.

We also observed that eliminating ‘background’ class benefits the proposed CSD, because the predominant ‘background’ class affects the consistency loss much. As a way of reducing the influence of the background and achieving improved performance, we propose the Background Elimination (BE) method which excludes boxes with high background probability in the computation of consistency loss.

CSD can be applied to both the single-stage detector such as SSD [50] and the two-stage detector such as R-FCN [14]. Various ablation studies have been performed showing the benefits of the proposed consistency losses for classification and localization. Also the effect of BE has been experimentally confirmed. Experimental results show that the proposed CSD improves the detection performance for all the detectors experimented.

Our main contributions can be summarized as follows:

- We propose a novel consistency-based semi-supervised learning algorithm for object detection that can be applied not only to single-stage

detectors but also to two-stage detectors.

- The proposed consistency constraints for object detection work well for both the classification of a bounding box and the regression of its location.
- We propose the BE method to mitigate the effect of background and show improvement of performance in most cases.

3.2 Method

The CSD to be presented works differently depending whether it is for a single-stage or for a two-stage object detector. The overall CSD structure for single-stage and two-stage object detectors is depicted in Fig. 3.2 and 3.3 respectively. The proposed structure is the combination of the Π -model in SSL [37] and an object detection algorithm. To allow one-to-one correspondence of target objects, an original image, I , and its flipped version, \hat{I} , are used as inputs. As in Fig. 3.2 and 3.3, a paired bounding box should represent the same class and their localization information must remain consistent.

During the training process, each mini-batch includes both labeled and unlabeled images. The labeled samples are trained using the typical object detection approach. The consistency loss is additionally applied to both the labeled and unlabeled images. In section 3.2.1, we explain the association method of corresponding boxes as well as the objective function used for training the object classifier in a single stage object detector. Likewise, in section 3.2.2, we define the objective function used for localization in both images. In the following sections afterward, we explain how these loss functions are utilized and show that our method is also applicable to a two-stage object detector.

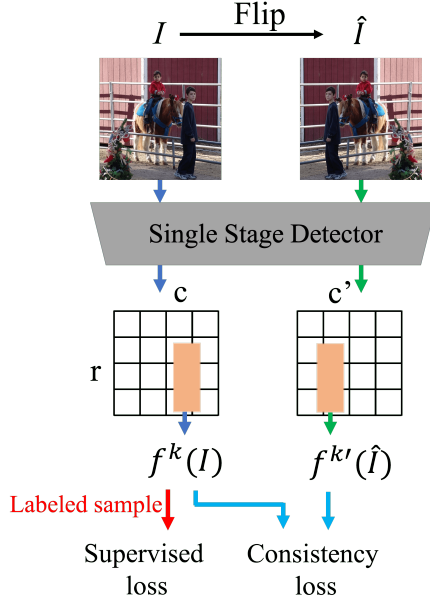


Figure 3.2: Overall structure of our proposed method for single stage detector. $f^k(I)$ and $f^{k'}(\hat{I})$ are extracted by a single stage detector from image I and flipped image \hat{I} respectively. The supervised loss is computed between $f^k(I)$ and the ground truth for labeled data and the consistency loss is computed between $f^k(I)$ and $f^{k'}(\hat{I})$ for labeled and unlabeled data.

3.2.1 Consistency loss for classification

We denote $f_{cls}^{p,r,c,d}(I)$ as the output class probability vector after softmax operation corresponding to the p -th pyramid, r -th row, c -th column and d -th default box. Since \hat{I} is a horizontally flipped version of I , predictions of two images should be equivalent. Also we want to make these vectors, $f_{cls}^{p,r,c,d}(I)$ and $f_{cls}^{p,r,c',d}(\hat{I})$, share a very similar distribution where $c' = C - c + 1$ and C is horizontal spatial dimension of the feature map. In semi-supervised learning, some

candidates such as L_2 distance or α -Jensen-Shannon divergence (α -JSD / for $\alpha = 1 \Rightarrow$ Jeffreys Divergence (JD)) can be used as the consistency regularization loss.

$$\begin{aligned} JS_\alpha &= KL(p||((1-\alpha)p + \alpha q)) + KL(q||((1-\alpha)q + \alpha p)) \\ JS_{\alpha=1} &= JD = KL(p||q) + KL(q||p) \end{aligned} \quad (3.1)$$

Among them, we specifically take advantage of JD for the following reasons. L_2 loss treats all the classes equal and in our case, consistency loss for irrelevant classes with low probability can affect the classification performance much. We experimentally observed that the performance of SSL with L_2 consistency loss is even worse than that of the supervised learning. To simplify the notation, we denote the location (p, r, c, d) as k and the horizontally opposite location (p, r, c', d) as k' . The classification consistency loss used for a pair of bounding boxes in our method is given as below:

$$l_{con_cls}(f_{cls}^k(I), f_{cls}^{k'}(\hat{I})) = JD(f_{cls}^k(I), f_{cls}^{k'}(\hat{I})) \quad (3.2)$$

where JD represents the Jeffreys Divergence. The overall consistency loss for classification is then obtained from the average of loss values from all bounding box pairs:

$$\mathcal{L}_{con-c} = \mathbb{E}_k[l_{con_cls}(f_{cls}^k(I), f_{cls}^{k'}(\hat{I}))] \quad (3.3)$$

3.2.2 Consistency loss for localization

The localization result for the k -th candidate box $f_{loc}^k(I)$ consists of $[\Delta cx, \Delta cy, \Delta w, \Delta h]$, which represent the displacement of the center and scale coefficients of a candidate box, respectively. Unlike the pair $(f_{cls}^k(I), f_{cls}^{k'}(\hat{I}))$, $f_{loc}^k(I)$ and $f_{loc}^{k'}(\hat{I})$ require a simple modification to be equivalent to each other. Since the

flipping transformation makes $\Delta \hat{c}x$ move in the opposite direction, a negation should be applied to correct it.

$$\begin{aligned}\Delta cx^k &\iff -\Delta \hat{c}x^{k'} \\ \Delta cy^k, \Delta w^k, \Delta h^k &\iff \Delta \hat{c}y^{k'}, \Delta \hat{w}^{k'}, \Delta \hat{h}^{k'}\end{aligned}$$

The localization consistency loss used for a single pair of bounding boxes in our method is given as below:

$$\begin{aligned}l_{con_loc}(f_{loc}^k(I), f_{loc}^{k'}(\hat{I})) &= \frac{1}{4}(\|\Delta cx^k - (-\Delta \hat{c}x^{k'})\|^2 + \|\Delta cy^k - \Delta \hat{c}y^{k'}\|^2 \\ &\quad + \|\Delta w^k - \Delta \hat{w}^{k'}\|^2 + \|\Delta h^k - \Delta \hat{h}^{k'}\|^2)\end{aligned}\tag{3.4}$$

The localization loss of each pair of bounding boxes and the total consistency loss are computed in the same principle as in the previous section:

$$\mathcal{L}_{con-l} = \mathbb{E}_k[l_{con_loc}(f_{loc}^k(I), f_{loc}^{k'}(\hat{I}))]\tag{3.5}$$

3.2.3 Overall loss for object detection

The total consistency loss is composed of the losses from section 3.2.1 and 3.2.2 as in

$$\mathcal{L}_{con} = \mathcal{L}_{con-c} + \mathcal{L}_{con-l}\tag{3.6}$$

Eventually, the final loss \mathcal{L} is composed of the original object detector's classification loss \mathcal{L}_c and localization loss \mathcal{L}_l , in addition to the consistency loss mentioned above. As in the typical semi-supervised learning methods [37, 71], ramp-up and ramp-down techniques, which can be defined by the weight scheduling $w(t)$, are used for the stable training.

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_l + w(t) \cdot \mathcal{L}_{con}\tag{3.7}$$

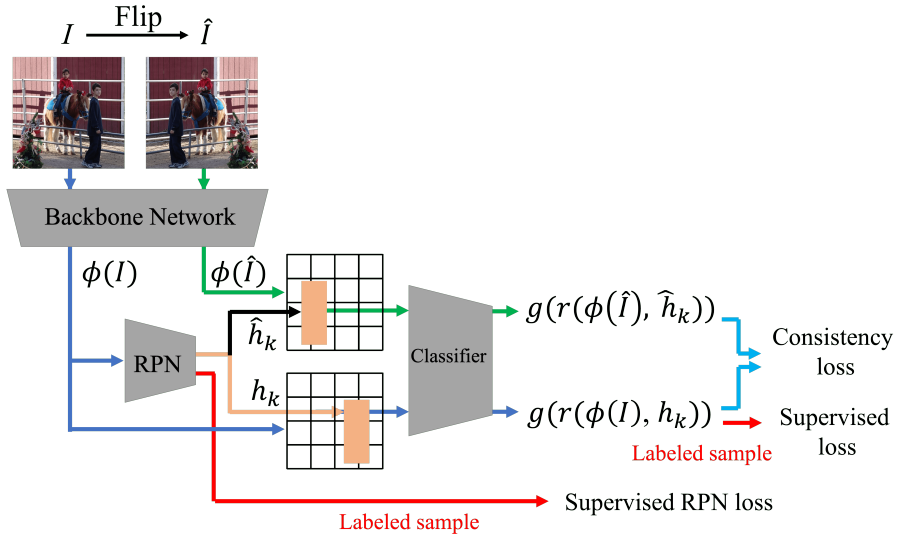


Figure 3.3: Overall structure of our proposed method for two stage detector. $\phi(I)$ and $\phi(\hat{I})$ originate from the backbone network and the RoI is computed only from $\phi(I)$. \hat{h}_k is obtained by flipping h_k to associate two corresponding boxes and supervised and consistency losses are calculated in the same way as for the single stage detector.

3.2.4 Application to two-stage detector

Unlike the single-stage detector, the two-stage detector has region proposal network (RPN) to generate region proposals and recognize the objectness of them. If we pass both the original and the flipped images to the RPN, the correspondence matching problem between the region proposals occurs which is relatively hard to solve. To simplify the problem, we only pass the feature $\phi(I)$ generated from the original image to the RPN. Then the output RoI locations are reversed and applied to the corresponding feature $\phi(\hat{I})$ as shown in Fig. 3.3. Given the

feature map $\phi(\hat{I})$ from the backbone network and the k -th RoI, h_k , from the RPN, the RoI-specific feature map of \hat{I} corresponding to the flipped area \hat{h}_k can be easily derived. As shown in Fig. 3.3 RPN is trained without the consistency loss. The features corresponding to the RoI, $r(\phi(I), h_k)$ and $r(\phi(\hat{I}), \hat{h}_k)$, are processed by a classifier g . Then, outputs $g(r(\phi(I), h_k))$ and $g(r(\phi(\hat{I}), \hat{h}_k))$ are used to compute the loss to train the network. As will be seen in the experiments, compared to the single-stage detector, the performance improvement of the proposed CSD is lower for two-stage detector and this attributes to the lack of consistency loss in RPN training.

3.2.5 Background Elimination

Particularly in object detection, an additional class of ‘background’ exists and most of the candidate boxes are usually classified to this class unless it is filtered by a confidence threshold. Consequently, consistency losses computed with all candidates will be easily dominated by backgrounds. This can degrade the classification performance for the foreground classes. Therefore, we exclude boxes having a high probability of background class by marking it with a mask. The mask is created according to the classification result for every candidate bounding box of I as in

$$m^k = \begin{cases} 1, & \text{if } \operatorname{argmax}(f_{cls}^k(I)) \neq \text{background} \\ 0, & \text{otherwise.} \end{cases} \quad (3.8)$$

Applying the mask to (3.3) and (3.5) yields

$$\begin{aligned} \mathcal{L}_{con-c} &= \mathbb{E}_{\mathbb{I}_{m^k=1}} [l_{con-cls}(f_{cls}^k(I), f_{cls}^{k'}(\hat{I}))], \\ \mathcal{L}_{con-l} &= \mathbb{E}_{\mathbb{I}_{m^k=1}} [l_{con-loc}(f_{loc}^k(I), f_{loc}^{k'}(\hat{I}))] \end{aligned} \quad (3.9)$$

where $\mathbb{I}_{m^k=1}$ indicates that the expectation is taken only for the positive mask. The overall process of the proposed CSD is described in Algorithm 1

3.3 Experiments

In our experiments, we have utilized the PASCAL VOC [23] and MSCOCO [49] datasets, as mention in 2.2.1. For PASCAL dataset, PASCAL VOC2007 trainval is used as the labeled data and PASCAL VOC2012 trainval and MSCOCO are utilized as the unlabeled one. We use test set of PASCAL VOC2007 for testing. For MS COCO dataset, we divided the MS COCO 2014 dataset into the existing categorized Train2014 (83k images) and Val2014-35k (35k images) dataset because minor classes may not be in the labeled dataset with random sampling. We trained our model with Val 35k dataset as labeled data and Train 83k as unlabeled data. Then, we tested with MS COCO test-dev dataset.

The codes used for our experiments are based on Pytorch. We have used third-party codes for SSD [50] ¹ and R-FCN [14] ². All experiments have been done under the similar setting with the code³ of the author. Expediently, labeled and unlabeled data are gathered in a single dataset and then randomly shuffled. In our setting, both labeled and unlabeled samples sit together in each mini-batch. The experimental settings of R-FCN are referred to those of SSM. As the batch size used for R-FCN is 4, using the same sampling strategy of SSD experiments does not guarantee that at least one labeled data is included in every mini-batch. To solve this problem, we have established separate data-loaders for labeled and unlabeled data. The amount of unlabeled data in a mini-batch is

¹<https://github.com/amdegroot/ssd.pytorch>

²<https://github.com/princewang1994/R-FCN.pytorch>

³<https://github.com/weiliu89/caffe/tree/ssd>

Algorithm 1 Training procedure of the proposed CSD

Require: $\mathcal{D}_{\mathcal{L}}, \mathcal{D}_{\mathcal{U}}$: labeled and unlabeled datasets

Require: $w(t)$: weight scheduling function

Require: $f(\cdot)$: trainable object detection model

Require: $h(\cdot)$: horizontal flip function

Require: $m(\cdot)$: objectness masks

- 1: **for** each $t \in [1, \text{max_iterations}]$ **do**
 - 2: ***Data Preparation***
 - 3: $\mathcal{A} \leftarrow \mathcal{D}_{\mathcal{L}} \cup \mathcal{D}_{\mathcal{U}}, \hat{\mathcal{A}} \leftarrow h(\mathcal{A})$
 - 4: ***Compute the outputs***
 - 5: $f(\mathcal{A}), f(\hat{\mathcal{A}})$
 - 6: ***Compute the objectness mask***
 - 7: $m_{\mathcal{A}} \leftarrow f(\mathcal{A})$ (Eq. 3.8)
 - 8: ***Compute the supervised & CSD losses***
 - 9: $\mathcal{L}_{\mathcal{S}} \leftarrow f(\mathcal{A} \in \mathcal{D}_{\mathcal{L}} \cap \mathcal{A})$
 - 10: $\mathcal{L}_{\text{CSD}} \leftarrow f(\mathcal{A} \in \mathcal{D}_{\mathcal{U}} \cap \mathcal{A}), f(\hat{\mathcal{A}}), m_{\mathcal{A}}$
 - 11: ***Compute the total loss***
 - 12: $\mathcal{L}_{\text{Total}} \leftarrow \mathcal{L}_{\mathcal{S}} + w(t) \cdot (\mathcal{L}_{\text{CSD}})$
 - 13: ***Update $f(\cdot)$ using $\mathcal{L}_{\text{Total}}$***
 - 14: **end for**
-

three times larger than that of the labeled data⁴. The total number of RoIs for CSD is 2k and all the parameter settings and training details are presented in the supplementary material.

3.3.1 Implementation Detail

Single-stage detector

Both SSD300 and SSD512 are used in our experiments. The authors of SSD provide the code in a github repository and we have followed the experimental settings in it. The backbone network is VGG16 pre-trained with ImageNet dataset. With PASCAL VOC dataset, models in all experiments have been trained for 120k iterations with a mini-batch size of 32. The learning rate is multiplied by 0.1 at 80k and 100k iterations. For the weight scheduling function $w(t)$, we have followed the policy of temporal ensembling [37]. The function is defined as below:

$$w(t) = \begin{cases} \exp^{-5 \times (1 - \frac{t}{t_1})^2}, & t < t_1 \\ 1, & t_1 \leq t < T - t_2 \\ \exp^{-12 \times (1 - \frac{T-t}{t_2})^2}, & t \geq T - t_2. \end{cases} \quad (3.10)$$

Here, T denotes the total number of iterations while t_1 and t_2 represent the ramp-up and ramp-down coefficients respectively. In experiments using PASCAL VOC dataset, t_1 is set to 32k and t_2 is set to 20k. Experiments of VOC-only COCO (COCO[†]) and Full COCO(COCO[§]) have been done with 240k and 360k of iterations, and parameter scheduling is also changed 2 ~ 3 times, ex-

⁴During the training, we allow the labeled data and unlabeled data not to share the same epoch number.

cluding ramp-down. If the ramp-down is longer, the influence of consistency is less affected by learning. Therefore, the ramp-down is maintained. A proper organization of a single mini-batch plays an important role for stable training. In case of COCO[§], the ratio of labeled data to unlabeled data is 1:26 which means that labeled data may be omitted in a mini-batch. To compensate this, unlabeled data are sampled according to the total number of samples in VOC12.

Two-stage detector

In experiments using a two-stage detector, we have adopted R-FCN under the same setting of SSM. ResNet-101 is used as a backbone network. In experiments using PASCAL VOC dataset, every model has been trained for 70k iterations with a batch size of 4. The learning rate is multiplied by 0.1 at 50k iterations. t_1 is set to 20k and t_2 is set to 10k. Experiments of VOC-only COCO (COCO[†]) and Full COCO(COCO[§]) have been done with 140k and 210k of iterations, and parameter scheduling is also changed 2 ~ 3 times, excluding ramp-down. In R-FCN, the ratio of positive regions to negative regions is 1:3 and this rate is applied to COCO as well.

3.3.2 Ablation Study

We have examined the influence of \mathcal{L}_{con-c} , \mathcal{L}_{con-l} and Background Elimination (BE) on SSD300, SSD512 and R-FCN and the performances are presented in Table 3.1 and 3.2. For SSD300, supervised learning using VOC07 and VOC0712 show 70.2 mAP and 77.2 mAP respectively as shown in Table 3.1. Using \mathcal{L}_{con-c} with Jeffreys Divergence induces 1.4% of improvement while \mathcal{L}_{con-c} with L_2 -norm causes a performance degradation to 70.0 mAP, which is slightly lower than that of the supervised learning. \mathcal{L}_{con-l} shows 2.0%

Table 3.1: Single-Stage Detection results for PASCAL VOC2007 test set. The first two rows show the performance of each detector by supervised learning. * is the score from [50, 14]. The following experiments use VOC07 as the labeled data and VOC12 as the unlabeled data, and show the results of the proposed CSD with/without \mathcal{L}_{con-c} (cls), \mathcal{L}_{con-l} (loc) and EB. The numbers in the parentheses are the performance enhancement over the baseline.

| Labeled data | Unlabeled data | Consistency | | Background | mAP (%) | |
|--------------|----------------|-------------|-----|-------------|-------------------|-------------------|
| | | cls | loc | Elimination | SSD 300 | SSD 512 |
| VOC07 | - | - | - | - | 68.0*/70.2 | 71.6*/73.3 |
| VOC0712 | - | - | - | - | 74.3*/77.2 | 76.8*/79.6 |
| VOC07 | VOC12 | ✓ | - | - | 71.6 (1.4) | 74.6 (1.3) |
| | | - | ✓ | - | 72.2 (2.0) | 74.6 (1.3) |
| | | ✓ | ✓ | - | 72.0 (1.8) | 74.8 (1.5) |
| VOC07 | VOC12 | ✓ | - | ✓ | 71.7 (1.5) | 75.4 (2.1) |
| | | - | ✓ | ✓ | 71.9 (1.7) | 75.2 (1.9) |
| | | ✓ | ✓ | ✓ | 72.3 (2.1) | 75.8 (2.5) |

of enhancement and jointly using both consistency losses shows 1.8% of enhancement. Particularly in SSD300 using \mathcal{L}_{con-l} only has shown better performance than using both. SSD512 scored 73.3 mAP and 79.6 mAP in pure supervised learning on VOC07 and VOC0712 respectively. Separate use of \mathcal{L}_{con-c} or \mathcal{L}_{con-l} induces 1.3% of improvements in both cases and joint usage of both losses improves 1.5% of accuracy. BE significantly improves the performance when used with both of the consistency losses. Especially, since more regions are predicted as backgrounds in SSD512 compared to SSD300, BE is more beneficial to SSD512 than to SSD300. Particularly in the case of using con-l in SSD300, the performance is quite good even without using BE. We think that it is because the number of samples with flat areas is relatively small for a small input resolution and this helps the regression process while con-c disturbs the effect of con-l.

As mentioned in section 3.2.4, CSD in R-FCN uses consistency losses only after the RoI pooling and not in the RPN. For R-FCN, supervised learning using VOC07 and VOC0712 shows 73.9 mAP and 79.4 mAP of accuracy respectively as shown in Table 3.2. There are small or no performance improvement before applying BE. However, when BE is applied, performance is improved by adding \mathcal{L}_{con-c} and \mathcal{L}_{con-l} . In addition, the performance is further improved with simultaneous use of both consistency losses.

3.3.3 Unlabeled data with different distribution (MSCOCO)

To see the effect of unlabeled data with different distribution to the labeled set, we use VOC07 as the labeled data and VOC12 plus MSCOCO as the unlabeled data as shown in Table 3.3. We denote ‘trainval’ of the MSCOCO dataset (123,287 images) as COCO[§] and the dataset (19,592 images) of which images

Table 3.2: Two-Stage Detection results for PASCAL VOC2007 test set. The first two rows show the performance of each detector by supervised learning. * is the score from [50, 14]. The following experiments use VOC07 as the labeled data and VOC12 as the unlabeled data, and show the results of the proposed CSD with/without \mathcal{L}_{con-c} (cls), \mathcal{L}_{con-l} (loc) and EB. The numbers in the parentheses are the performance enhancement over the baseline.

| Labeled data | Unlabeled data | Consistency | | Background | mAP (%) |
|--------------|----------------|-------------|-----|-------------|-------------------|
| | | cls | loc | Elimination | R-FCN |
| VOC07 | - | - | - | - | 73.9 |
| VOC0712 | - | - | - | - | 79.5*/79.4 |
| VOC07 | VOC12 | ✓ | - | - | 74.0 (0.1) |
| | | - | ✓ | - | 73.9 (0.0) |
| | | ✓ | ✓ | - | 74.0 (0.1) |
| VOC07 | VOC12 | ✓ | - | ✓ | 74.5 (0.6) |
| | | - | ✓ | ✓ | 74.4 (0.5) |
| | | ✓ | ✓ | ✓ | 74.7 (0.8) |

Table 3.3: Detection results on PASCAL VOC2007 test set. “COCO[§]”: All 80 classes. “COCO[†]”: 20 PASCAL VOC classes.

| Labeled data | Unlabeled data | CSD Method (mAP) | | |
|--------------|-------------------------|------------------|-------------|-------------|
| | | SSD300 | SSD512 | R-FCN |
| VOC07 | - | 70.2 | 73.3 | 73.9 |
| VOC07 | VOC12 | 72.3 | 75.8 | 74.7 |
| | VOC12+COCO [§] | 71.7 | 75.1 | 74.9 |
| | VOC12+COCO [†] | 72.6 | 75.9 | 75.1 |

contain only objects belonging to the 20 PASCAL VOC classes as COCO[†].

In single-stage detectors, the performance by training with unlabeled VOC12 and COCO[§] shows better performance than the supervised learning, but it is less than the performance using unlabeled VOC12 only. In a two stage detector, it shows higher performance than the supervised learning and training with unlabeled VOC12 data. Training with unlabeled VOC12 and COCO[†], both the single-stage detector and two-stage detector show performance improvements. We analyze this phenomenon in the next section.

3.3.4 MSCOCO

Table 3.4 shows the results of experiments on the MSCOCO dataset. The supervised performances of SSD using Val35k and Trainval35k show 18.8 mAP and 23.9 mAP, respectively. CSD with Val35k labeled data and Train80k unlabeled data on SSD shows 1.0% of enhancement.

Table 3.4: Detection results for MS COCO test-dev set. The following experiments use Val35k (labeled) and Train80k (unlabeled) data. The numbers in the parentheses are the performance improvements from the baseline model (SSD trained on Val35k). All experiments are tested by ourselves.

| Method | Labeled data | Unlabeled data | Avg. Precision, IoU: | | |
|------------|---------------------------------|----------------|----------------------|------------|------------|
| | | | 0.5:0.95 | 0.5 | 0.75 |
| SSD300 | Val35k | - | 18.8 | 34.8 | 18.6 |
| | Val35k + Train80k (trainval35k) | - | 23.9 | 40.8 | 24.7 |
| Ours (CSD) | Val 35k | Train 80k | 19.8 (1.0) | 35.8 (1.0) | 19.8 (1.2) |

3.4 Discussion

3.4.1 Consistency regularization with only labeled data

We evaluated our method on PASCAL VOC 2007 under the supervised training setting. We observed that training with the consistency loss only on labeled data led to worse results. It means that the consistency loss does not affect the improvement of performance for labeled data. We conjecture this problem as follows. A model trained with a small amount of labeled data can be easily fitted, and the classification probability of output of the objects will have very high confidence ($f_{cls}^k(I) \geq 0.9$). At this time, if the consistency loss is applied, the fitting outputs ($f_{cls}^k(I)$ and $f_{cls}^{k'}(I)$) are computed as each other’s target ($f_{cls}^{k'}(I)$, and $f_{cls}^k(I)$) and it may cause overfitting. It shows the same tendency in conventional semi-supervised learning [37, 71]. On the other hand, in the case of semi-supervised setting, the classification probability of output of objects is no longer sharp because of unlabeled data. Therefore, our consistency constraints are helpful to improve the performance for semi-supervised object detection

Table 3.5: Detection results for PASCAL VOC2007 test set. The first two rows show the performance of each detector by supervised learning. * is the score from [50]. The following experiments use VOC07 as the labeled data and show the results of the proposed CSD with/without \mathcal{L}_{con-c} (cls), \mathcal{L}_{con-l} (loc) and BE.

| Labeled data | Unlabeled data | Consistency | | Background Elimination | Method (mAP) |
|--------------|----------------|-------------|-----|------------------------|--------------|
| | | cls | loc | | |
| VOC07 | - | - | - | - | 68.0*/70.2 |
| VOC0712 | - | - | - | - | 74.3*/77.2 |
| VOC07 | - | ✓ | - | - | 69.4 |
| | | - | ✓ | - | 69.9 |
| | | ✓ | ✓ | - | 69.7 |
| VOC07 | - | ✓ | - | ✓ | 70.2 |
| | | - | ✓ | ✓ | 69.8 |
| | | ✓ | ✓ | ✓ | 69.3 |

Table 3.6: Effects of using Background Elimination (BE) on SSD300 performance.

| VOC07(L)+VOC12(U) | mAP |
|------------------------------|------|
| without BE | 72.0 |
| BE with m^k | 72.3 |
| BE with $m^k \otimes m^{k'}$ | 71.7 |

task.

3.4.2 Single-stage detector vs. Two-stage detector:

We apply consistency constraint differently depending on whether RPN is used or not. First, in a single-stage detector, the proposed consistency losses can be applied to all areas and it shows much improvement in performance. The two-stage detector, on the other hand, uses \hat{h} by flipping the h obtained from I . Therefore, while we can expect to improve performance in the classifier, it is hard to expect additional performance improvement of RPN. As a result, the two-stage detector has less performance improvement than the single-stage detector. To optimize the RPN, a new way exploiting the consistency loss is needed, which we leave as further work.

3.4.3 Background Elimination:

The proportion of background in the predefined boxes is very large. We apply BE to reduce the effect of the background and show that BE is helpful in improving the performance. However, getting rid of too many samples is not helpful in

learning, as shown in Table 3.6. As a way of reducing more background samples, the consistency losses are applied to the candidate boxes only when their estimated class is non-background ($m^k = 1$) as well as their flipped boxes on the flipped images are estimated as non-background ($m^{k'} = 1$). At this time, the performance of the SSD300 model shows 71.7 mAP, which is 0.6% lower than the original 72.3 mAP. This shows that removing too many background samples may cause performance degradation.

3.4.4 Datasets

Table 3.3 shows that in learning 20 classes of VOC, additional usage of unlabeled data leads to an enhanced performance. However, the ratio of labeled/unlabeled class mismatch decides the amount of improvement. This is why the case of using VOC12 + COCO[†] shows a better result than the case of VOC12 + COCO[§]. This result is consistent with the recent study by [55].

BE is hardly expected to remove this out-of-distribution. It is intended to eliminate background, but classes in MSCOCO can have a higher confidence in other classes that are similar. For example, classes such as ‘giraffe’ and ‘elephant’ may have features similar to ‘horse’ or ‘dog’ rather than the background. These data can interfere with training detectors.

On the other hand, adding unlabeled data with a similar distribution, all detectors have improved the performance. Our CSD does not need any labeling in the additional data but it still has its limitation that the distribution of the unlabeled data should be similar to that of the labeled data. Further research is needed to solve this problem, which we leave it for future work.

Table 3.7: Comparisons between self-training and consistency regularization based methods on PASCAL VOC2007 test set. “COCO[§]”: All 80 classes. “COCO[†]”: 20 PASCAL VOC classes.

| Single-Stage Detector | | | | |
|------------------------|--------------|---------------------------|-------------|------------|
| Method | Labeled data | Unlabeled data | mAP | Gain |
| SSD512 (supervised) | VOC07 | - | 73.3 | - |
| SSD512 + CSD (ours) | VOC07 | VOC12 | 75.8 | 2.5 |
| SSD512 + CSD (ours) | VOC07 | VOC12 + COCO [§] | 75.1 | 1.8 |
| SSD512 + CSD (ours) | VOC07 | VOC12 + COCO [†] | 75.9 | 2.6 |
| Two-Stage Detector | | | | |
| Method | Labeled data | Unlabeled data | mAP | Gain |
| R-FCN (supervised) | VOC07 | - | 73.9 | - |
| RFCN + SPL (300%) [35] | VOC07 | VOC12 + COCO [§] | 74.1 | 0.2 |
| RFCN + SPL (400%) [35] | VOC07 | | 74.7 | 0.8 |
| RFCN + SSM (300%) [82] | VOC07 | | 75.6 | 1.7 |
| RFCN + SSM (400%) [82] | VOC07 | | 76.7 | 2.8 |
| RFCN + CSD (ours) | VOC07 | VOC12 | 74.7 | 0.8 |
| RFCN + CSD (ours) | VOC07 | VOC12 + COCO [§] | 74.9 | 1.0 |
| RFCN + CSD (ours) | VOC07 | VOC12 + COCO [†] | 75.1 | 1.2 |

3.4.5 Self-training vs. Consistency regularization

Self-training is widely used as a simple heuristic method in semi-supervised learning. As it is an iterative method which cycles training, prediction of unlabeled data and changing the training dataset, it is time-consuming and computationally intensive [62]. In addition, the threshold and stop criterion, which decide the quality and quantity of an additional dataset, affect the algorithm's performance. Meanwhile, CR method which trains unlabeled data with an additional loss helps the more common and robust learning.

Table 3.7 shows the performance of SPL [35], SSM [82] and CSD. SPL and SSM are based on the self-training method, which shows different performance depending on the amount of data added⁵. They experimented only with R-FCN framework and trained with VOC07 as labeled data and VOC12 and MSCOCO as unlabeled data. The performance of SPL is improved by 0.2 and 0.8 than baseline while SSM has 1.7 and 2.8 better performance than the baseline. In CSD, according to unlabeled dataset, it shows performance improvement of 0.8 ~ 1.2 than baseline. As mentioned above, CSD has a limitation in the two stage detector, which has less performance improvement than single stage detector. In single stage detector, however, SSD512 shows the 1.8% and 2.6% performance improvements.

Comparison with SSM: For a practical usage of a model, the stopping criteria must be defined properly. SSM continues the training until no more unlabeled data are included in the training set (Algorithm 1 in [82]). In Fig.4 of [82], the performance improves initially as unlabeled samples are added but it starts to degrade as more samples are added. However, the reported score of the SSM is not

⁵% means that the percentage of additional unlabeled objects over labeled objects.

from an objective stopping criterion but the peak performance during the entire iterations. With this setting, the influx of the data from out-of-distribution and incorrectly labeled samples cannot be prevented. As we all know that the performance of SSM should not be measured with this setting, we are not sure that the performance of a detector combined with SSM would get better. To check this, we tried to implement SSM in SSD. However, many details are missing in SSM and the learning parameters of single-stage detector and two-stage detector are different. Its intense time-cost and huge hyper-parameter space makes it difficult to implement SSM properly. In our work, we just wanted to present a representative self-learning method.

3.5 Conclusion

We have introduced a novel Consistency-based Semi-supervised learning for object Detection (CSD) method. To the best of our knowledge, it is the first attempt to extend CR used in conventional semi-supervised classification problems to object detection problem. We applied the proposed CSD to single-stage detectors and a two-stage detector respectively and designed loss to improve the performance of both detectors over the supervised learning method. We have shown that consistency loss is helpful for semi-supervised learning in classification as well as localization with various ablation experiments. In addition, BE has been shown to improve performance.

Chapter 4

Interpolation-based Semi-supervised learning for object Detection (ISD)

4.1 Introduction

In this chapter, we address the semi-supervised object detection problem and propose a new method called Interpolation-based Semi-supervised learning for object Detection (ISD) whose loss terms can also be applied to the supervised learning framework. Interpolation Regularization (IR) which mixes different images and learns to predict the combined label rather than one hot vector performs outstandingly in supervised learning as well as in semi-supervised learning for classification problems [89, 75, 76, 5, 78].

However, it is challenging to apply IR directly to object detection because the background class consists of a very diverse and irregular texture. Fig. 4.1 shows an example of applying IR to the object detection problem. As shown in Fig. 4.1, we mix image A and B using the mixing parameter $\lambda = 0.5$ as shown in the middle. Obviously, the mixed green box has 50% of a dog and 50% of a bird as we can see in Fig. 4.2(a). However, when an object is mixed

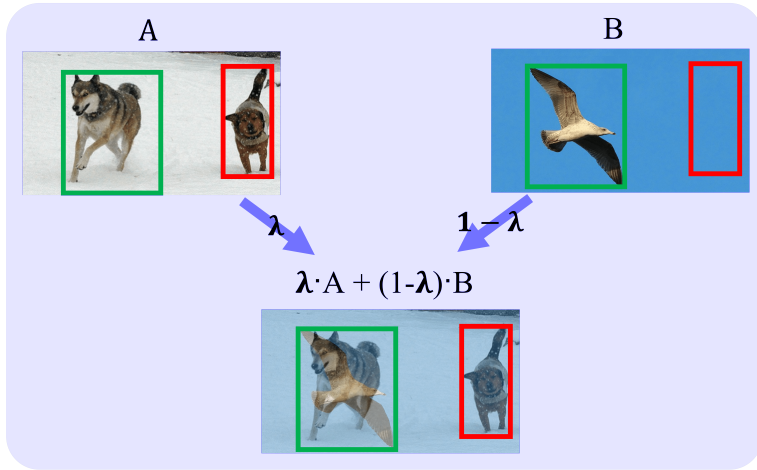
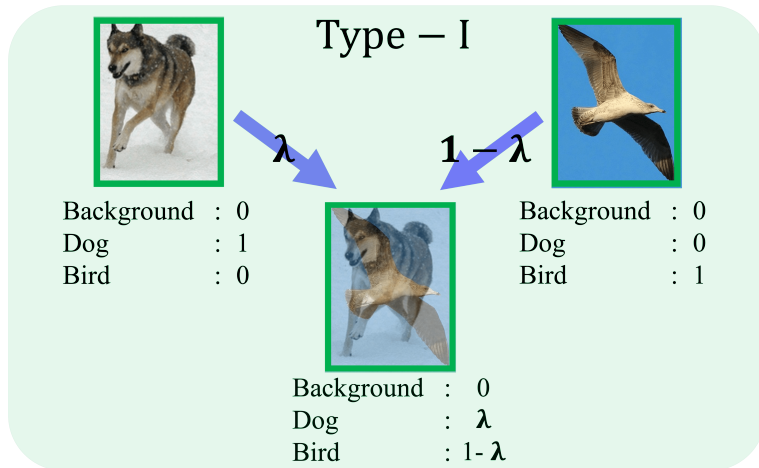


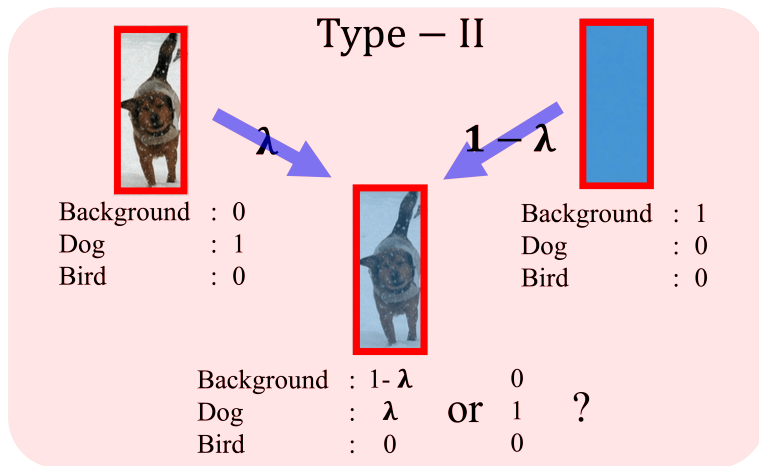
Figure 4.1: Mixed image created by random interpolation between images A and B

with a background as in Fig. 4.2(b), the mixed image appears to be an 100% object corrupted by noise. If the detector is trained by the conventional IR, any blurred or noisy mixture images contribute to increasing the confidence of the background class, and it will degrade performance. On the other hand, if that sample is trained as a foreground object, it is expected to be robust to noise and to learn about various backgrounds around the object.

To tackle this problem, in this chapter, we divide the mixed images into two types (Type-I and II) depending on whether one of the original images is the background or not. Then, we apply a different IR algorithm suitable for each type. The proposed ISD method which will be detailed in next chapter can be combined with conventional semi-supervised learning methods such as CSD (consistency-based semi-supervised learning) to improve the semi-supervised object detection performances. Also, the proposed scheme can be used to enhance the detection performance in the supervised learning framework. Our



(a)



(b)

Figure 4.2: (a) Type-I : both patches are from object classes. (b) Type-II : one of the patches is from the object class.

main contributions can be summarized as follows:

- We show the problem in applying interpolation regularization directly to the object detection task and propose a novel interpolation-based semi-supervised learning algorithm for object detection.
- In doing so, we define two types of interpolation in the object detection task and propose efficient semi-supervised learning methods suitable for each type.
- We experimentally show the effectiveness of the proposed method for each type by demonstrating a significant performance improvement over the conventional semi-supervised object detection algorithms.

4.2 Method

We denote an image created by random mixing, $\lambda \cdot A + (1 - \lambda) \cdot B$, of two images A and B as $Mix_\lambda(A, B)$.

$$Mix_\lambda(A, B) = \lambda \cdot A + (1 - \lambda) \cdot B \quad (4.1)$$

Similar to Mixup, the mixing coefficient λ is drawn from the $Beta(\alpha, \alpha)$ distribution. In our method, we use SSD [50], one of the most popular single-stage object detectors, as a baseline detector. The network output of SSD $f^{p,r,c,d}$ is denoted as the output of the p^{th} layer of the pyramid, r^{th} row, c^{th} column and d^{th} default box, and (p, r, c, d) is expressed as k for brevity. Each f^k is composed of f_{cls}^k and f_{loc}^k which are the softmax output vector and the localization offsets of the center and the size of the box, $[\Delta cx, \Delta cy, \Delta w, \Delta h]$, at position

k , respectively. The mask $m(I)$, which is computed by $f_{cls}(I)$, is used in background elimination and interpolation type categorization for image I and has the binary objectness value at each location k :

$$m(I)^k = \begin{cases} 1, & \text{if } \operatorname{argmax}(f_{cls}^k(I)) \neq \text{background} \\ 0, & \text{otherwise.} \end{cases} \quad (4.2)$$

4.2.1 Type categorization.

We determine the type of a pair of patches by the background elimination method that only extracts patches with a high objectness probability. Then we apply different methods appropriate for each type of patches. Eq. (4.3) is how we calculate each type of a mask. The Type-I mask, m_I , is calculated by element-wise multiplication of $m(A)$ and $m(B)$. In other words, it becomes 1 when both patches of $m(A)^k$ and $m(B)^k$ are 1, and otherwise it is 0. On the other hand, the Type-II mask m_{II}^A is calculated by element-wise multiplication of $m(A)$ and $\sim m(B)$, which means it is 1 when the patch in image A has a high objectness score while the corresponding patch at the same location in image B has a high background score, and vice versa for m_{II}^B .

$$\begin{aligned} \text{Type-I mask: } m_I &= m(A) \otimes m(B), \\ \text{Type-II(A) mask: } m_{II}^A &= m(A) \otimes \sim m(B), \\ \text{Type-II(B) mask: } m_{II}^B &= \sim m(A) \otimes m(B). \end{aligned} \quad (4.3)$$

4.2.2 Type I loss

When the patches in image A and B are all likely to be objects (Type-I), we define a Type-I loss inspired by the ICT loss [76]. Note that there are two dif-

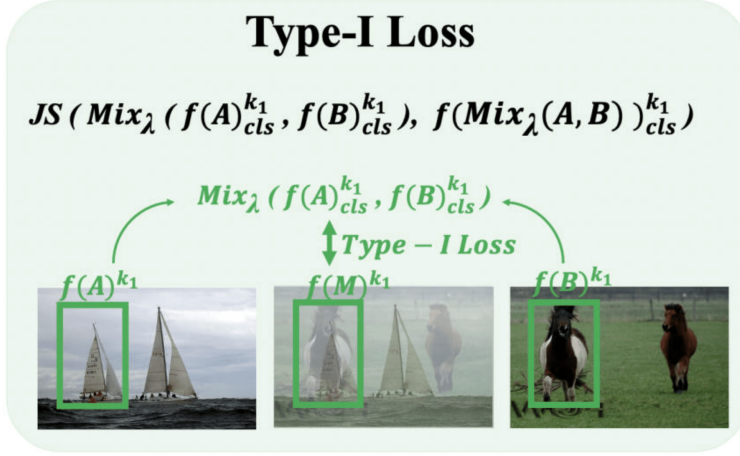


Figure 4.3: Type-I Loss : both patches are from object classes

ferences compared to the conventional ICT. First, we used α -Jensen-Shannon divergence (α -JSD / for $\alpha = 1 \Rightarrow$ Jeffreys Divergence (JD)) as the consistency regularization loss (function $d(.,.)$ in Eq. (2.2)).

$$JS_{\alpha} = KL(p||((1 - \alpha)p + \alpha q) + KL(q||((1 - \alpha)q + \alpha p)) \quad (4.4)$$

$$JS_{\alpha=1} = JD = KL(p||q) + KL(q||p)$$

In the CSD, JD shows better performance because L2 loss equally weights all the classes, including the background class. Second, we use the same network to feed-forward inputs like CSD, distinct from ICT which uses different networks for mixed and original inputs using MeanTeacher [71]. Eq. (4.5) shows the loss function of Type-I, which is the distance between the mixed output of $f(A)_{cls}^k$ and $f(B)_{cls}^k$ and the output of the mixed image of A and B, $f(Mix_{\lambda}(A, B))_{cls}^k$.

$$l_I = JD(Mix_{\lambda}(f(A)_{cls}^k, f(B)_{cls}^k)||f(Mix_{\lambda}(A, B))_{cls}^k) \quad (4.5)$$

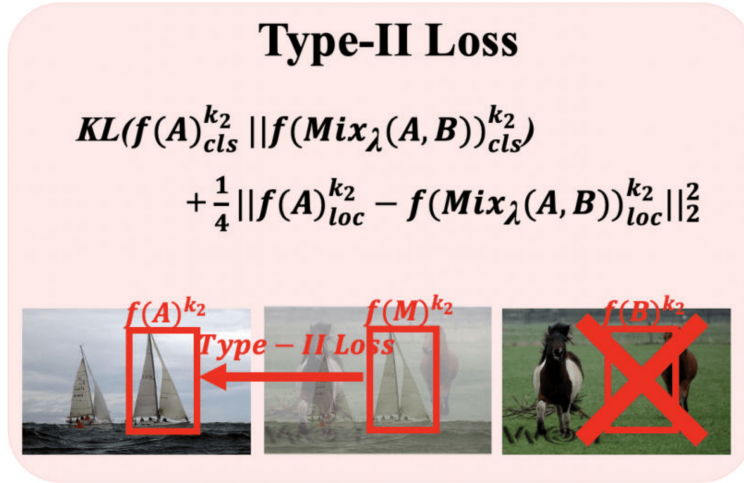


Figure 4.4: Type-II : one of the patches is from the object class

The overall Type-I loss \mathcal{L}_I is the average of patches whose Type-I mask is

1.

$$\mathcal{L}_I = \mathbb{E}_{\mathbb{I}\{m_I=1\}}[l_I]. \quad (4.6)$$

Here, \mathbb{E} and \mathbb{I} are the expectation and the indicator function, respectively.

4.2.3 Type II loss

As shown in Fig. 4.4, in Type II, one patch has a high probability of foreground, while the other has a high probability of background. In this case, instead of using the Type I loss described above, we train the network to have similar predictions on the mixed patch and the patch with a high probability of foreground. This kind of loss can be interpreted as a form of FixMatch loss [67] which encourages consistency between the predictions on the strong augmen-

tation and the weak augmentation of an input. More specifically, in our case, the mixed patch is considered as a strong augmentation while the patch with a high foreground probability acts as no-augmentation. Note that, for classification, FixMatch is trained with targets by creating pseudo-labels of samples that exceed a threshold, whereas we do not need to set a specific threshold and the target is set according to the output distribution of no-augmentation patch.

We set $f(A)$ or $f(B)$ as a target, and train the mixed output ($f(\text{Mix}_\lambda(A, B))$) to be close to $f(A)$ or $f(B)$. In doing so, Kullback-Leibler (KL) divergence and L2 loss are used as the classification and localization losses, respectively as follows:

$$l_{II.cls}^A = KL(f(A)_{cls}^k || f(\text{Mix}_\lambda(A, B))_{cls}^k) \quad (4.7)$$

$$l_{II.loc}^A = \frac{1}{4} || f(A)_{loc}^k - f(\text{Mix}_\lambda(A, B))_{loc}^k ||_2^2. \quad (4.8)$$

The overall Type-II loss when patch A is foreground, \mathcal{L}_{II}^A , is calculated as the average of the sum of two individual losses as follows:

$$\mathcal{L}_{II}^A = \mathbb{E}_{\mathbb{1}\{m_{II}^A=1\}} [l_{II.cls}^A + l_{II.loc}^A]. \quad (4.9)$$

Likewise, \mathcal{L}_{II}^B is also calculated by applying the above loss, and the total loss of Type-II is calculated as follows:

$$\mathcal{L}_{II} = \mathcal{L}_{II}^A + \mathcal{L}_{II}^B. \quad (4.10)$$

Finally, the overall ISD loss is computed by Type-I loss (\mathcal{L}_I) and Type-II loss (\mathcal{L}_{II}) as follows:

$$\mathcal{L}_{ISD} = \gamma_1 \cdot \mathcal{L}_I + \gamma_2 \cdot \mathcal{L}_{II}. \quad (4.11)$$

Here, γ_1 and γ_2 are set appropriately to balance both loss terms. The overall process of the proposed semi-supervised learning is described in Algorithm 2.

Algorithm 2 Training procedure of the proposed ISD

Require: $\mathcal{D}_{\mathcal{L}}, \mathcal{D}_{\mathcal{U}}$: labeled and unlabeled datasets

Require: $w(t)$: weight scheduling function

Require: $f(\cdot)$: trainable object detection model

Require: $h(\cdot)$: horizontal flip function

Require: $m(\cdot)$: objectness masks

- 1: **for** each $t \in [1, \text{max_iterations}]$ **do**
 - 2: **Data Preparation**
 - 3: $\mathcal{A}, \mathcal{B} \leftarrow \mathcal{D}_{\mathcal{L}} \cup \mathcal{D}_{\mathcal{U}}$
 - 4: $\mathcal{C} \leftarrow \text{Mix}_{\lambda}(\mathcal{A}, \mathcal{B})$
 - 5: **Compute the outputs**
 - 6: $f(\mathcal{A}), f(\mathcal{B}), f(\mathcal{C})$
 - 7: **Compute the objectness mask**
 - 8: $m_{\mathcal{A}} \leftarrow f(\mathcal{A}), \quad m_{\mathcal{B}} \leftarrow f(\mathcal{B})$ (Eq. 4.2)
 - 9: **Compute the supervised loss**
 - 10: $\mathcal{L}_S \leftarrow f(\mathcal{A} \in \mathcal{D}_{\mathcal{L}} \cap \mathcal{A})$
 - 11: **Compute the ISD loss using the type mask (Eq. 4.3)**
 - 12: $\mathcal{L}_I \leftarrow \mathbb{E}_{\mathbb{I}\{m_I=1\}}[l_I]$ (Eq. 4.5)
 - 13: $\mathcal{L}_{II}^A \leftarrow \mathbb{E}_{\mathbb{I}\{m_{II}^A=1\}}[l_{II}^A_{cls} + l_{II}^A_{loc}]$ (Eq. 4.7, 4.8)
 - 14: $\mathcal{L}_{II}^B \leftarrow \mathbb{E}_{\mathbb{I}\{m_{II}^B=1\}}[l_{II}^B_{cls} + l_{II}^B_{loc}]$
 - 15: $\mathcal{L}_{II} \leftarrow \mathcal{L}_{II}^A + \mathcal{L}_{II}^B$
 - 16: $\mathcal{L}_{ISD} \leftarrow \lambda_1 \cdot \mathcal{L}_I + \lambda_2 \cdot \mathcal{L}_{II}$
 - 17: **Compute the total loss**
 - 18: $\mathcal{L}_{Total} \leftarrow \mathcal{L}_S + w(t) \cdot (\mathcal{L}_{ISD})$
 - 19: **Update** $f(\cdot)$ **using** \mathcal{L}_{Total}
 - 20: **end for**
-

4.3 Experiments

All experiments have been done under the similar setting with the experimental settings chapter 3. Similar to CSD, we experimented on the PASCAL VOC dataset and MS COCO dataset with SSD300 model. VGG-16 pre-trained model is used as our backbone network. For VOC dataset, we followed the settings from the conventional Semi-Supervised Learning methods for object detection. Similar to [82] and CSD, we trained our model with PASCAL VOC07 *trainval* (5k images) dataset as labeled data and PASCAL VOC12 *trainval* (12k images) as unlabeled data. Then, we tested with PASCAL VOC07 test dataset. For MSCOCO dataset, we divided the MSCOCO 2014 dataset into the existing categorized Train2014 (83k images) and Val2014-35k (35k images) dataset because minor classes may not be in the labeled dataset with random sampling. We trained our model with Val 35k dataset as labeled data and Train 83k as unlabeled data. Then, we tested with MS COCO test-dev dataset.

We sample the mixing parameter λ from $Beta(\alpha, \alpha)$ at every iteration. The parameters are set to $(\gamma_1, \gamma_2) = (0.1, 1)$ in Eq. (4.11) and $\alpha = 100$ in the beta distribution. Other learning parameters such as the learning rate and the batch size are the same as CSD.

4.3.1 PASCAL VOC

Supervised Learning

We start by examining the effect of ISD on SSD in the supervised training setting, i.e, the proposed losses in 4.11 are applied to labeled data. The results are presented in Table 4.1. In the first row block, SSD (base) trained with VOC 07 (*trainval*) data shows 70.2 mAP performance, while that of SSD (CSD) de-

Table 4.1: Detection results for PASCAL VOC2007 test set under the supervised training setting. L_{cls} and L_{loc} are the consistency classification and localization loss with BE in CSD. The following experiments use VOC07 (labeled) and VOC12 (unlabeled) data. **Blue** and **Red** are represented as the Baseline score and Best score, respectively. The numbers in the parentheses are the performance increments compared with the baseline.

| Semi-Supervised Loss | Labeled data | Unlabeled data | mAP (%) |
|--|---------------|----------------|-------------|
| Supervised Learning – Trained only with labeled data | | | |
| None [50] | VOC07 | - | 70.2 |
| (Supervised Learning) | VOC07 + VOC12 | - | 77.2 |
| CSD | VOC07 | - | 69.3 |
| Ours (ISD only) | VOC07 | - | 72.3 |

creases to 69.3 mAP, which shows a side effect that we mentioned in 3.4.1. On the other hand, SSD300 (ISD) shows 2.1% improvement in accuracy compared to SSD (base).

Semi-Supervised Learning

We evaluate the performance of ISD in the Semi-Supervised Learning setting. As shown in Table 4.2, the performance of the SSD model trained only with VOC07 labeled data is 70.2%. Type-I and Type-II show 1.7% and 3.6% of enhancement, respectively. The Type-I consists of only classification loss, and it shows better result than the score of only classification loss in CSD. Type-II shows much better performance than CSD and jointly using both Type-I and Type-II losses shows 3.9% of enhancement. This demonstrates the effectiveness

Table 4.2: Detection results for PASCAL VOC2007 test set under the semi-supervised training setting. L_{cls} and L_{loc} are the consistency classification and localization loss with BE in CSD. The following experiments use VOC07 (labeled) and VOC12 (unlabeled) data. **Blue** and **Red** are represented as the Baseline score and Best score, respectively. The numbers in the parentheses are the performance increments compared with the baseline.

| Semi-Supervised Loss | Labeled data | Unlabeled data | mAP (%) |
|-----------------------------|---------------|----------------|-------------------|
| None [50] | VOC07 | - | 70.2 |
| (Supervised Learning) | VOC07 + VOC12 | - | 77.2 |
| Semi-Supervised Learning | | | |
| CSD (L_{cls}) | | | 71.7 (1.5) |
| CSD (L_{loc}) | VOC07 | VOC12 | 71.9 (1.7) |
| CSD ($L_{cls} + L_{loc}$) | | | 72.3 (2.1) |
| ISD (Type-I only) | | | 71.9 (1.7) |
| ISD (Type-II only) | VOC07 | VOC12 | 73.8 (3.6) |
| ISD (Type-I,II) | | | 74.1 (3.9) |

of our approach in the SSL setting.

4.4 Discussion

4.4.1 Ablation studies for Type-I and Type-II losses

We experiment to verify the performance of the two types of loss we proposed in Table 4.2. Each loss shows a significant performance improvement compared to the supervised learning. In the table, for all the cases, the Type-II loss performed better than of Type-I loss. There are three reasons for this results. First, the numbers of Type-I and Type-II samples are different. With a trained model, the number of Type-II samples was 5 times that of Type-I samples, which indicates that the influence of Type-I loss is relatively small. Second, Type-I only considers the classification loss while Type-II uses the localization loss as well. Because the two objects in Type-I have different bounding boxes, the boundary of their mixed patch is not equal to the interpolation of their bounding boxes. Therefore, the localization loss cannot be applied in Type-I cases. Third, two objects that are mixed may not be aligned well. More research is needed for the alignment in Interpolation Regularization, which remains as a future work.

In Table 4.3¹, we analyzed the effect of the classification and the localization loss in Type-II when α is 100. The classification loss on Type-II samples showed more remarkable performance improvement than the localization loss, and by combining them, we can obtain better performance.

In the case of Type-II, it shows a very large performance improvement compared to Type-I. And we can think of a way to make strong augmented data

¹Note that the model is combined with the CSD, and the combining method will be described in the next chapter.

Table 4.3: Ablation study of Type-II losses on PASCAL VOC2007 test set. All the experiments in this table are performed by adding each loss to the CSD. (α is 100).

| VOC07(L)+VOC12(U) | mAP (%) |
|---------------------|-------------|
| Type-II (cls) | 74.0 |
| Type-II (loc) | 73.1 |
| Type-II (cls + loc) | 74.2 |

like mixed images by generating images to fit object detection. However, the purpose of this paper is to inform that there is a problem when applying interpolation regularization and to suggest a way to solve it. In addition, the above mentioned method [68] was published by the google brain team at the same time as our paper and shows good performance.

4.4.2 Beta distribution

In ISD, the mixing coefficient λ is sampled from the $Beta(\alpha, \alpha)$ distribution. Table 4.4² shows the performance of ISD using various values of α across different types of ISD losses. We observe that a large range of α gives improved performance in comparison to the baseline (CSD with 72.3% mAP). In general, we recommend to set α to a sufficiently large value. The reason for choosing relatively large α is as follows: With a smaller values of α (e.g. $\alpha < 1$), λ will be close to either 0 or 1 with high probability, thus most of the mixed images will be closer to either of the original images being mixed. In this case,

²Note that the model is combined with the CSD, and the combining method will be described in the next chapter.

Table 4.4: Ablation study for α and each type in VOC07(L) + VOC12(U) training dataset and VOC07 testing dataset. The row represents the α of the beta distribution, and the column represents each type. All the experiments in this table are performed by adding each loss to the CSD.

| $\beta(\alpha, \alpha)$ | SSD300 + ISD Method (mAP (%)) | | |
|-------------------------|-------------------------------|-------------|------------------|
| α | Type-I | Type-II | Type-I + Type-II |
| 1 | 72.3 | 72.8 | 72.9 |
| 10 | 72.4 | 73.8 | 74.0 |
| 100 | 72.4 | 74.2 | 74.4 |
| 1000 | 72.2 | 74.2 | 74.3 |

the mixed image M will be extremely weak (for one image) or strong (for the other) augmentation resulting in lowered performance with high variance. In contrast, increasing the values of α increases the probability of λ being closer to 0.5, which provides an appropriate level of regularization. Note that if the value of α is too large, λ will be concentrated too much around 0.5 and all the augmented samples will be too different from the original images resulting in degraded performance with high variance at test time.

4.4.3 Training model size

For ISD training, image batches are inferred by the network three times over conventional SSD. Also, due to the calculation of additional losses, it requires more than three times the conventional SSD memory. We used Nvidia 1080Ti GPU, and we assigned 4 GPUs for SSD model with ISD training. With fewer GPUs, our implementation was not trainable because of limited memory budget.

However, at testing, it has the same network size and inference time as the base network and can improve the performance.

4.4.4 Object detector

In this chapter, we have used the SSD model among various single stage detectors. In the case of other detectors, algorithm-specific modifications should be made to successfully apply interpolation regularization, while the basic idea of separating Type-I and Type-II samples and applying a different loss for each case is still valid. In the case of a Two-Stage detector, generally, Region of Interest (RoI) is obtained by Region Proposal Network (RPN) and classification of that location is performed for object detection. Since the RoIs of A , \hat{A} , B , and $Mix_\lambda(A, B)$ are all different, in order to apply our algorithm, one of RoIs should be applied to other images for one-to-one correspondence. If the RoI of A is applied to other images, Type-II loss between B and $Mix_\lambda(A, B)$ cannot be obtained, and if each RoI of A , B , $Mix_\lambda(A, B)$ is applied individually to other images, a lot of computation will be required. Thus how to apply interpolation-based regularizer for Two-stage detectors is an interesting avenue for further research.

4.5 Conclusion

In this chapter, we have proposed ISD, a simple and efficient Interpolation-based semi-supervised learning algorithm for object detection using single-stage detectors. We started by investigating the challenges that occur when the Interpolation Regularization methods for the classification task are applied directly to an object detection task, and have addressed these challenges by proposing dif-

ferent types of interpolation-based loss functions. Our method shows significant improvement in both semi-supervised and supervised object detection tasks over the previous methods, compared over the same dataset and the same architecture settings. We leave the exploration of Interpolation Regularization for Two-stage detectors as a future work.

Chapter 5

Combination of CSD and ISD

In this chapter, we introduce the method of combining CSD and ISD. As state in chapter 3, CSD not only computes the original batch but also predicts another batch that is horizontally flipped images. Therefore, CSD requires twice as much memory as conventional training. On the other hand, as mentioned in chapter 4, ISD needs predictions of the original mini-batch, other new mini-batch, and another mini-batch that are mixed between the original mini-batch and other. Therefore, ISD allocates three times as much memory as conventional training. Since the combination of CSD and ISD needs four types of predictions mentioned above, it requires four times as much memory as conventional training. However, even the SSD300 model, which requires a small amount of memory, cannot be trained with four 1080ti GPUs at the same batch size.

We proposed a method of using horizontal flip image batch in CSD to efficiently use memory and easily combine the two algorithms. However, it is difficult to apply the horizontal flip image batch directly. The mixed image $Mix_\lambda(A, \hat{A})$ of $A \in \mathcal{A}$ and its horizontal flipped version $\hat{A} \in \hat{\mathcal{A}}$ would have similar backgrounds and predict the same class in the center of the image, as

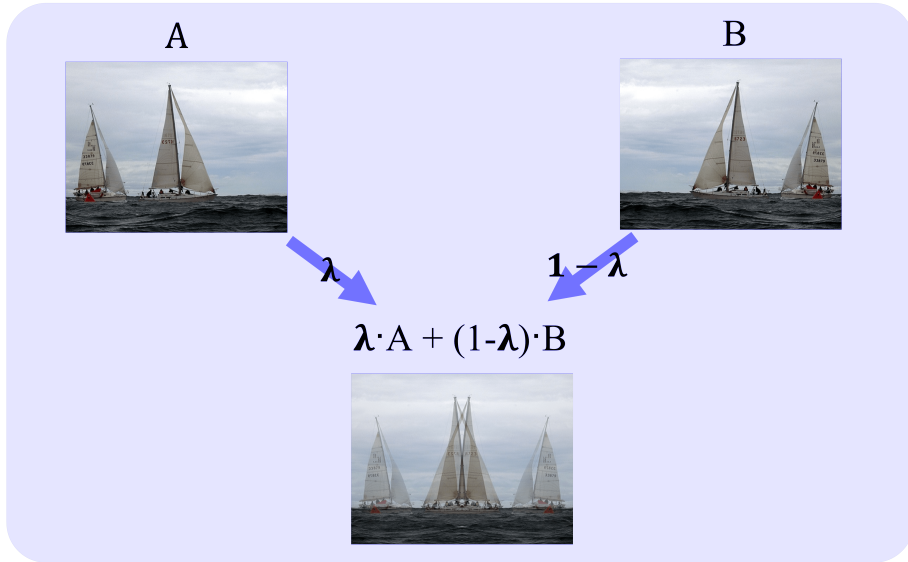


Figure 5.1: The mixed image $Mix_{\lambda}(A, \hat{A})$ of $A \in \mathcal{A}$ and its horizontally flipped version $\hat{A} \in \hat{\mathcal{A}}$

shown in Fig. 5.1. Therefore, by applying shuffle and mixing in batch, we got the same result as a prediction of 4 times with a prediction of 3 times. Our main contributions can be summarized as follows:

- We show the problem in mixing the original image and the horizontally flipped image and propose a method to combine CSD and ISD with less memory by shuffle and mix mini-batch.
- We experimentally show the effectiveness of combining CSD and ISD by verifying a significant performance improvement over the previous CSD and ISD algorithms.

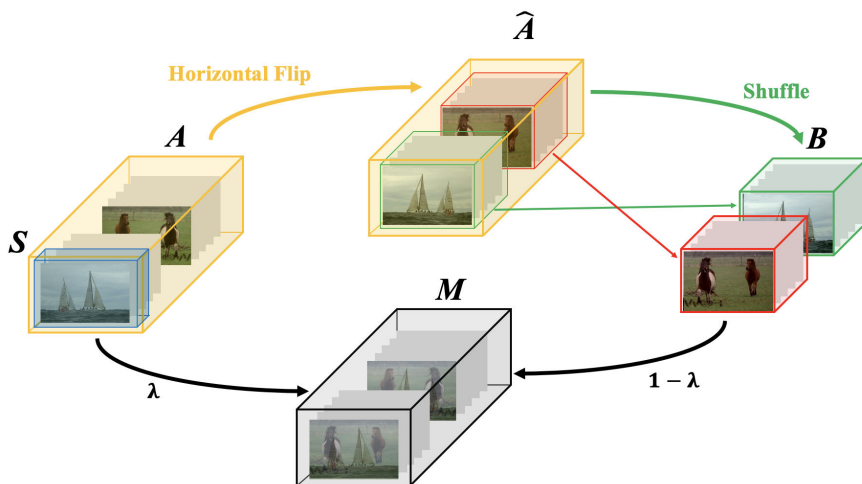


Figure 5.2: Combination of ISD with CSD. The original images (\mathcal{A}) are flipped ($\hat{\mathcal{A}}$) and the mixed images (\mathcal{M}) are obtained by combining the two. First, the order of flipped images are changed by shuffling ($\mathcal{B} = \text{shuffle}(\hat{\mathcal{A}})$), then \mathcal{A} and \mathcal{B} are mixed ($\mathcal{M} = \text{Mix}_{\lambda}(\mathcal{A}, \mathcal{B})$). CSD loss is calculated between \mathcal{A} and $\hat{\mathcal{A}}$ and ISD loss is computed between \mathcal{M} and (\mathcal{A} and/or \mathcal{B}). In the original set (\mathcal{A}), the blue box (\mathcal{S}) is labeled, to which the supervised loss is applied.

5.1 Method

For ISD training, three sets of image batches, \mathcal{A} , \mathcal{B} , and $\mathcal{M} = \text{Mix}_\lambda(\mathcal{A}, \mathcal{B})$ should be inferred by the network. For efficient training, we set \mathcal{B} as the horizontally flipped version of \mathcal{A} , i.e., $\hat{\mathcal{A}} = \text{flip}(\mathcal{A})$, as shown in Fig. 5.2. We calculated the CSD loss with those two batches. As shown in Fig. 5.2, we make the mixed images by combining the original batch (\mathcal{A}) with the half-shuffled flipped batch ($\mathcal{B} = \text{shuffle}(\hat{\mathcal{A}})$). The total loss consists of supervised loss (\mathcal{L}_S), CSD loss (\mathcal{L}_{CSD}), and ISD loss (\mathcal{L}_{ISD}) as follows:

$$\mathcal{L}_{Total} = \mathcal{L}_S + w(t) \cdot [\mathcal{L}_{CSD} + \mathcal{L}_{ISD}], \quad (5.1)$$

where $w(t)$ is a weight scheduling function. The overall process of the proposed semi-supervised learning is described in Algorithm 3

5.2 Experiments

All experiments have been done under the similar setting with the experimental settings chapter 3 and 4.

5.2.1 PASCAL VOC

Supervised Learning

We start by examining the effect of combining CSD and ISD on SSD in the supervised training setting, i.e., the proposed losses in 5.1 are applied to labeled data. The results are presented in Table 5.1. SSD (base) trained with VOC 07 (*trainval*) data shows 70.2 mAP performance, while that of SSD (CSD) decreases to 69.3 mAP. On the other hand, SSD300 (ISD) and SSD (ISD + CSD)

Algorithm 3 Training procedure of the proposed Combination of CSD with ISD

Require: $\mathcal{D}_{\mathcal{L}}, \mathcal{D}_{\mathcal{U}}$: labeled and unlabeled datasets

Require: $w(t)$: weight scheduling function

Require: $f(\cdot)$: trainable object detection model

Require: $h(\cdot)$: horizontal flip function

Require: $m(\cdot)$: objectness masks

- 1: **for** each $t \in [1, \text{max_iterations}]$ **do**
 - 2: **Data Preparation**
 - 3: $\mathcal{A} \leftarrow \mathcal{D}_{\mathcal{L}} \cup \mathcal{D}_{\mathcal{U}}, \hat{\mathcal{A}} \leftarrow h(\mathcal{A})$
 - 4: $\mathcal{B} \leftarrow \text{shuf}f\text{le}(\hat{\mathcal{A}})$
 - 5: $\mathcal{C} \leftarrow \text{Mix}_{\lambda}(\mathcal{A}, \mathcal{B})$
 - 6: **Compute the outputs**
 - 7: $f(\mathcal{A}), f(\hat{\mathcal{A}}), f(\mathcal{C})$
 - 8: $f(\mathcal{B}) \leftarrow \text{shuf}f\text{le}(f(\hat{\mathcal{A}}))$
 - 9: **Compute the objectness mask**
 - 10: $m_{\mathcal{A}} \leftarrow f(\mathcal{A}), m_{\mathcal{B}} \leftarrow f(\mathcal{B})$ (Eq. 4.2)
 - 11: **Compute the supervised & CSD losses**
 - 12: $\mathcal{L}_S \leftarrow f(\mathcal{A} \in \mathcal{D}_{\mathcal{L}} \cap \mathcal{A})$
 - 13: $\mathcal{L}_{CSD} \leftarrow f(\mathcal{A} \in \mathcal{D}_{\mathcal{U}} \cap \mathcal{A}), f(\hat{\mathcal{A}}), m_{\mathcal{A}}$
 - 14: **Compute the ISD loss using the type mask (Eq. 4.3)**
 - 15: $\mathcal{L}_I \leftarrow \mathbb{E}_{\mathbb{I}\{m_I=1\}}[l_I]$ (Eq. 4.5)
 - 16: $\mathcal{L}_{II}^A \leftarrow \mathbb{E}_{\mathbb{I}\{m_{II}^A=1\}}[l_{II}^A \text{.cls} + l_{II}^A \text{.loc}]$ (Eq. 4.7, 4.8)
 - 17: $\mathcal{L}_{II}^B \leftarrow \mathbb{E}_{\mathbb{I}\{m_{II}^B=1\}}[l_{II}^B \text{.cls} + l_{II}^B \text{.loc}]$
 - 18: $\mathcal{L}_{II} \leftarrow \mathcal{L}_{II}^A + \mathcal{L}_{II}^B$
 - 19: $\mathcal{L}_{ISD} \leftarrow \lambda_1 \cdot \mathcal{L}_I + \lambda_2 \cdot \mathcal{L}_{II}$
 - 20: **Compute the total loss**
 - 21: $\mathcal{L}_{Total} \leftarrow \mathcal{L}_S + w(t) \cdot (\mathcal{L}_{CSD} + \mathcal{L}_{ISD})$
 - 22: **Update** $f(\cdot)$ **using** \mathcal{L}_{Total}
 - 23: **end for**
-

Table 5.1: Detection results for PASCAL VOC2007 test set under the supervised training setting. L_{cls} and L_{loc} are the consistency classification and localization loss with BE in CSD. The following experiments use VOC07 (labeled) and VOC12 (unlabeled) data. **Blue** and **Red** are represented as the Baseline score and Best score, respectively. The numbers in the parentheses are the performance increments compared with the baseline.

| Semi-Supervised Loss | Labeled data | Unlabeled data | mAP (%) |
|--|---------------|----------------|-------------|
| Supervised Learning – Trained only with labeled data | | | |
| None [50] | VOC07 | - | 70.2 |
| (Supervised Learning) | VOC07 + VOC12 | - | 77.2 |
| CSD | VOC07 | - | 69.3 |
| ISD | | - | 72.3 |
| CSD + ISD | | - | 73.1 |

show 2.1% and 2.9% improvements in accuracy compared to SSD (base), respectively. This shows that combining ISD with a strong CSD regularizer stabilizes the training, making the network more robust.

Semi-Supervised Learning

We evaluate the performance of combining CSD and ISD in the semi-supervised learning setting. As shown in Table 5.2, the performance of the SSD model trained only with VOC07 labeled data is 70.2%. When CSD and ISD are combined, it shows even greater performance improvement. This demonstrates the effectiveness of our approach in the SSL setting. Moreover, ISD+CSD with VOC07 labeled data and VOC12 unlabeled data on SSD (Table 5.2, last row)

Table 5.2: Detection results for PASCAL VOC2007 test set under the semi-supervised training setting. L_{cls} and L_{loc} are the consistency classification and localization loss with BE in CSD. The following experiments use VOC07 (labeled) and VOC12 (unlabeled) data. **Blue** and **Red** are represented as the Baseline score and Best score, respectively. The numbers in the parentheses are the performance increments compared with the baseline.

| Semi-Supervised Loss | Labeled data | Unlabeled data | mAP (%) |
|---------------------------------|---------------|----------------|-------------------|
| None [50] | VOC07 | - | 70.2 |
| (Supervised Learning) | VOC07 + VOC12 | - | 77.2 |
| Semi-Supervised Learning | | | |
| CSD (L_{cls}) | | | 71.7 (1.5) |
| CSD (L_{loc}) | VOC07 | VOC12 | 71.9 (1.7) |
| CSD ($L_{cls} + L_{loc}$) | | | 72.3 (2.1) |
| ISD (Type-I only) | | | 71.9 (1.7) |
| ISD (Type-II only) | | | 73.8 (3.6) |
| ISD (Type-I,II) | VOC07 | VOC12 | 74.1 (3.9) |
| CSD+ISD | | | 74.4 (4.2) |

Table 5.3: Detection results for PASCAL VOC2007 test set. The following experiments use VOC07 (labeled) and VOC12 & MSCOCO (unlabeled) data.

| Detector | Labeled data | Unlabeled data | mAP (%) | |
|----------|--------------|-----------------------|---------|-------------|
| | | | CSD | ISD + CSD |
| SSD300 | VOC07 | VOC12 | 72.3 | 74.4 |
| | | VOC12 + MSCOCO (full) | 71.7 | 73.7 |
| | | VOC12 + MSCOCO(VOC) | 72.6 | 74.5 |

shows 1.3% performance improvement in comparison to the fully supervised setting with VOC07 labeled data on SSD (Table 5.1, last row). This explains that the combined loss of ISD+CSD not only on labeled data, but also on unlabeled data contributes to better performance. The results shown in Table 5.2 demonstrate that our ISD+CSD approach outperforms the baseline CSD-only approach by a significant margin.

5.2.2 Unlabeled data with different distribution (MSCOCO)

Similar to 3.3.3, we experimented by adding the MSCOCO dataset that has different distributions to the unlabeled data. In our results, as shown in Table 5.3 When VOC 12 + MSCOCO (VOC) is trained as unlabeled data, it shows better performance than VOC 12 alone as unlabeled data. On the other hand, when VOC12 + MSCOCO (full) is used as unlabeled data, the performance is deteriorated. This shows one of the semi-supervised learning limitations [55, 28], which can degrade performance as the out-of-class distribution data in the unlabeled dataset increases.

Table 5.4: Detection results for MS COCO test-dev set. The following experiments use Val35k (labeled) and Train80k (unlabeled) data. The numbers in the parentheses are the performance improvements from the baseline model (SSD trained on Val35k). All experiments are tested by ourselves.

| Method | Labeled data | Unlabeled data | Avg. Precision, IoU: | | |
|-----------|---------------------------------|----------------|----------------------|-------------------|-------------------|
| | | | 0.5:0.95 | 0.5 | 0.75 |
| SSD300 | Val35k | - | 18.8 | 34.8 | 18.6 |
| | Val35k + Train80k (trainval35k) | - | 23.9 | 40.8 | 24.7 |
| CSD | Val 35k | Train 80k | 19.8 (1.0) | 35.8 (1.0) | 19.8 (1.2) |
| CSD + ISD | | | 21.0 (2.2) | 37.7 (2.9) | 21.1 (2.5) |

5.2.3 MSCOCO

Table 5.4 shows the results of experiments on the MSCOCO dataset. The supervised performances of SSD using Val35k and Trainval35k show 18.8 mAP and 23.9 mAP, respectively. While CSD with Val35k labeled data and Train80k unlabeled data on SSD shows 1.0% of enhancement, CSD+ISD shows 2.1% performance improvement in the same experimental setting for the COCO dataset.

5.3 Discussion

5.3.1 CSD and ISD with only labeled data

We evaluated combination of CSD and ISD on PASCAL VOC 2007 under the supervised training setting. When CSD and ISD are combined together, it results in even better performance compared to the case only ISD is applied to the labeled data which also improves performance. On the contrary, applying only CSD to the labeled data decreases improvement. We analyzed this result as fol-

Table 5.5: Detection results for PASCAL VOC 2007 set. The following experiments use three types of VOC07 (labeled/ half-train, train, and trainval) and VOC12 (unlabeled) data.

| Labeled | SSD | Unlabeled | CSD | CSD + ISD |
|-------------------------|------|----------------------|------|-----------|
| VOC07 half-train (1.3k) | 41.5 | VOC12 trainval (12k) | 45.8 | 54.9 |
| VOC07 train (2.5k) | 63.6 | VOC12 trainval (12k) | 64.9 | 68.5 |
| VOC07 trainval (5k) | 70.2 | VOC12 trainval (12k) | 72.3 | 74.4 |

lows. In case of CSD, the supervised and the consistency losses are small values, so there is little change in the weight of the model. Therefore, both the original image and the flipped image have the fitted values, which can cause the overfitting as mentioned in 3.4.1. On the other hand, in the case of ISD, the model is trained with both Type-I and Type-II losses for training on the mixed image, so it cannot be easily fitted. In this case, the ISD loss has a huge value, which causes a lot of weight change in the model and prevents overfitting. Therefore, it is helpful to combine ISD with the CSD for enhanced result.

5.3.2 Small labeled dataset

Table 5.5 is the experimental results of each algorithm that is trained using the train dataset in VOC07, which consists of a various number of images (1.3k, 2.5k, and 5k). As the number of labeled data decreased, the performance of the SSD decreased significantly. At this time, the performance of CSD and CSD+ISD showed better performance than the existing supervised learning. Nevertheless, the results of CSD+ISD are inferior to those trained with a little more labeling data. In the case of CSD + ISD trained with VOC07 half-train

labeled data and VOC12 unlabeled data, a total of 13.3k data are used. It shows the 54.9% mAP score. At this time, the SSD trained only with the VOC 07 train label shows 63.6% performance, and the performance of CSD+ISD (VOC07 half-train labeled data and VOC12 unlabeled data) shows worse than supervised learning with VOC07 train labeled dataset.

5.3.3 Training model size

For CSD and ISD algorithms, we have to predict one and two additional mini-batches for computing the losses, respectively. Therefore, combining CSD with ISD is needed three additional prediction. In other words, since conventional SSD allocates 12 GB memory, proposed method requires 48GB memory. However, we have 4 1080Ti GPUs, and our total memory in GPU is 44 GB. In our method, as we shuffled the mini-batch output of horizontal flipped batches, we reduce the network prediction and memory.

5.4 Conclusion

In this chapter, we have introduced combining method of CSD and ISD. In combining the above two algorithms directly, we cannot train the model due to the out-of-memory. In order to utilize existing resources, we made the mixed image using mixing original image and horizontal flipped image. However, this method generates data so that it has a problem with the conventional training of ISD. Therefore, we adopted the shuffle and mix method so that results are equivalent to the method that predicts all mini-batch. We confirmed that the proposed algorithm shows significant improvement in not only supervised learning but also semi-supervised learning, and this algorithm shows improved performance

not only in the VOC dataset but also in the COCO dataset.

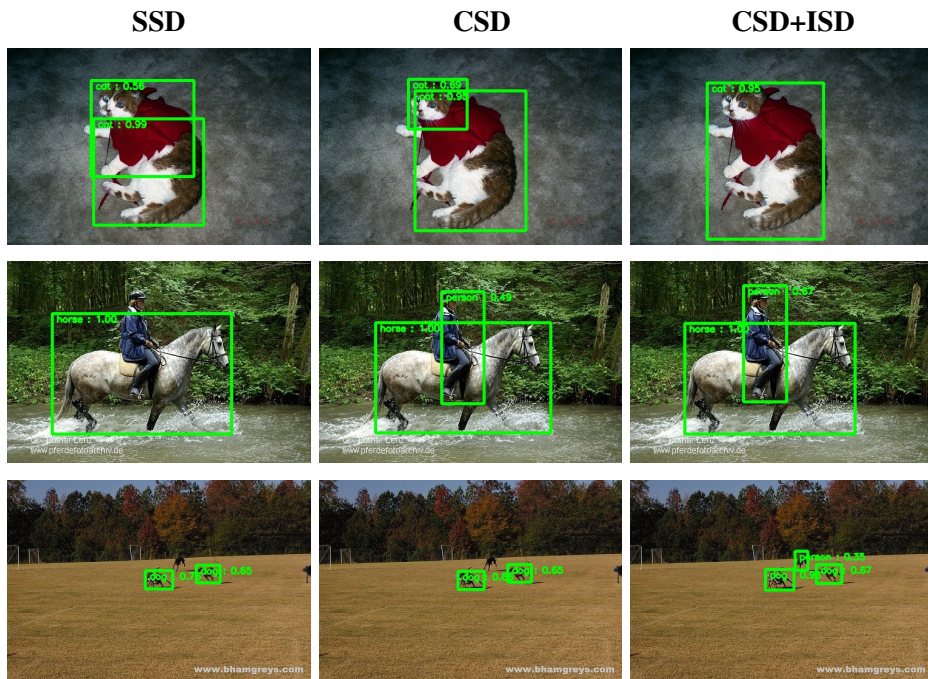


Figure 5.3: Qualitative results for the PASCAL VOC2007 test set using supervised SSD, semi-supervised CSD and CSD+ISD models in table 5.2. The first, middle, and last columns are the resulting images of the SSD, CSD, and CSD+ISD models, respectively. A score threshold of 0.3 is used to display these images.

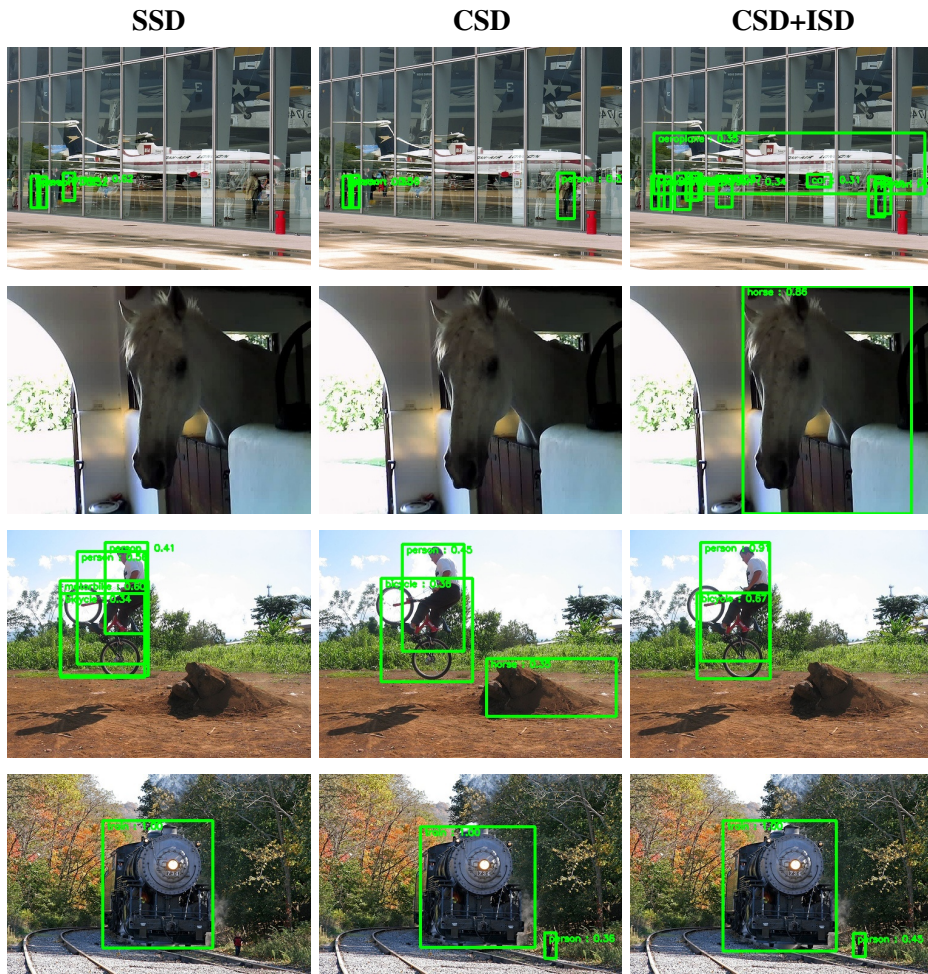


Figure 5.4: Qualitative results for the PASCAL VOC2007 test set using supervised SSD, semi-supervised CSD and CSD+ISD models in table 5.2. .

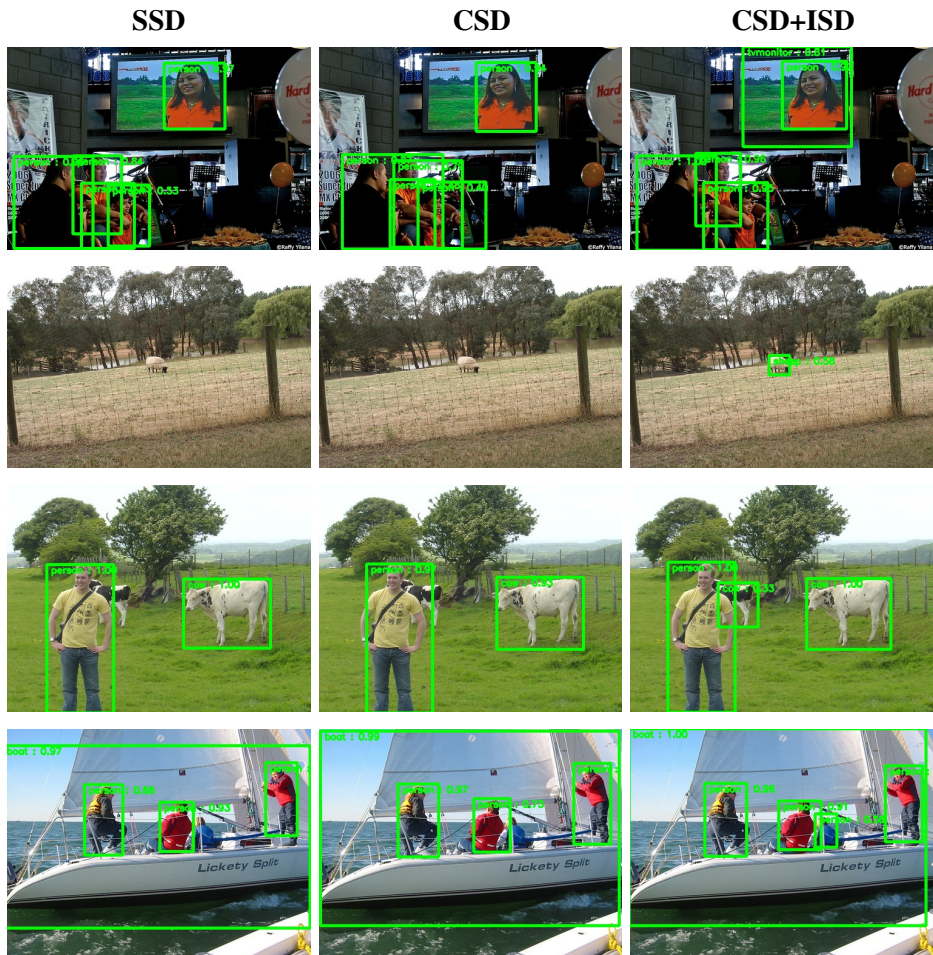


Figure 5.5: Qualitative results for the PASCAL VOC2007 test set using supervised SSD, semi-supervised CSD and CSD+ISD models in table 5.2.

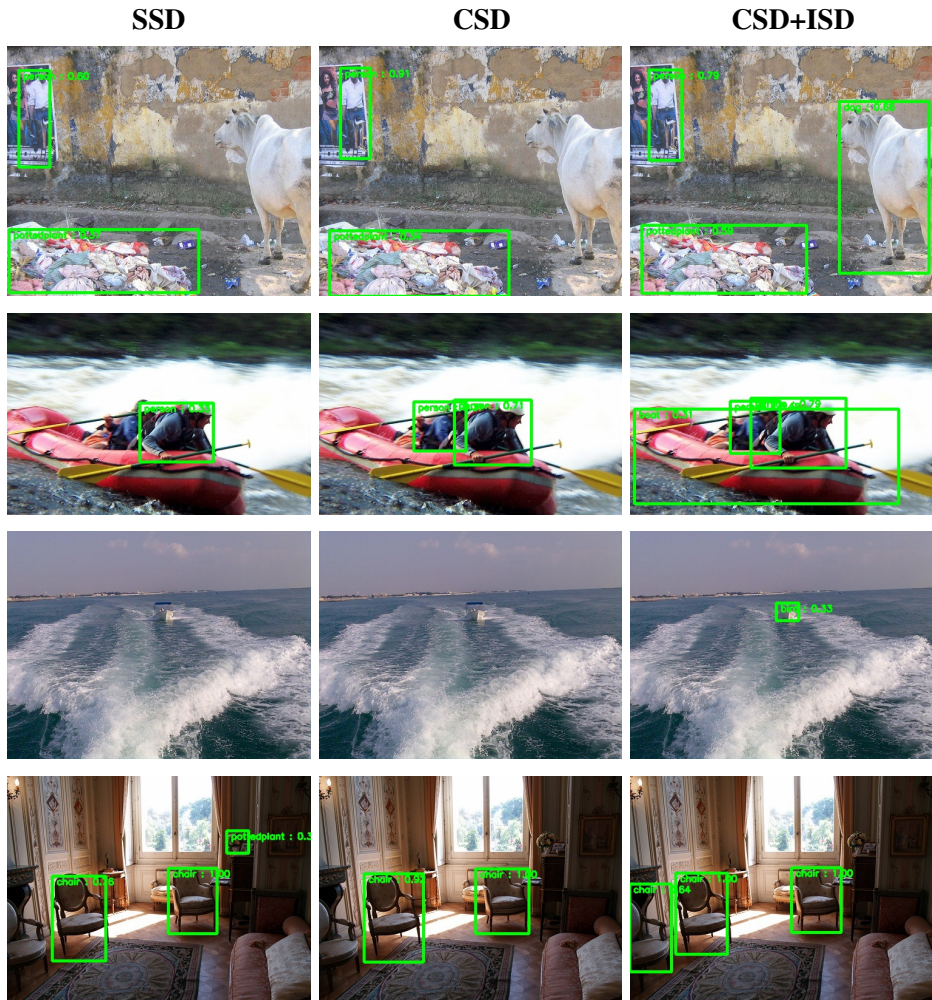


Figure 5.6: Qualitative results for the PASCAL VOC2007 test set using supervised SSD, semi-supervised CSD and CSD+ISD models in table 5.2.

Chapter 6

Conclusion

In this dissertation, we proposed various Semi-Supervised Learning methods for object detection. Prior to our proposed algorithms, algorithms that applied the self-training method are introduced, which take much time to train and are sensitive to hyperparameter. Our algorithms are the first attempt to extend CR and IR to object detection, which was only used in conventional semi-supervised classification problems. Also, we conducted research to find problems resulting from applying CR and IR to object detection at the same time. Finally, we solved the defined problem and achieved better result. Our proposed algorithms show great performance improvement compared to the supervised learning method and it is an important step in semi-supervised object detection in the future.

In this chapter, we give a brief summary of the proposed method and discuss the limitations along with future directions of our research.

6.1 Summary

Methods for Semi-Supervised Learning for object Detection are proposed throughout this dissertation, which improves the performance of object detectors by training labeled and unlabeled data together.

First, we proposed a way of applying Semi-Supervised Learning of the Consistency Regularization method to object detection. This is the first algorithm to apply a consistency-based semi-supervised classification problem to object detection. We have defined a new consistency loss for classification as well as localization and have verified that our algorithm improves performance in not only single-stage detector but also the two-stage detector. In addition, it was confirmed that the proposed Background Elimination alleviates the imbalance problem of the background and helps to improve performance.

Second, we also proposed a way of applying Semi-Supervised Learning of the Interpolation Regularization method to object detection. This is also the first algorithm to apply an interpolation-based semi-supervised classification problem to object detection. We discovered that directly applying it causes The IR problem and found a solution to deal with it which is reversely utilizing the problem. By adopting this method, the performance is improved.

Lastly, we proposed an algorithm that can utilize CSD and ISD at the same time. We faced the memory issue that occurs when using both of them at the same time. Therefore, we shuffle the flipped batch to use memory efficiently and achieve the same result. Memory problem is effectively solved enabling to train CSD and ISD simultaneously resulting in higher efficiency compared to each training performance.

6.2 Limitations

In this dissertation, there are three significant limitations:

First, the two-stage detector has structural limitations on applying to our algorithm. In a Two-stage Detector, Region Proposal Network (RPN) is used to extract Region of Interest (RoI), and classification is performed on these RoIs. At this time, the RPN is sampling the patches that are likely to be objects. Since the object candidates can be sampled from different locations under the minor perturbation in the same image, it is difficult to match the outputs. In the case of CSD, the outputs of RPN cannot be matched one-to-one correspondence for the same reason. Consequentially, the two-stage detector shows less improvement than the single-stage detector. In the case of ISD, more outputs are predicted than CSD, and matching the outputs is a more difficult problem.

Second, the proposed algorithms have a limitation in that they require a lot of memory. The proposed algorithm requires 2-3 times more memory than the conventional algorithm, making it challenging to apply to the latest object detector. SOTA object detectors, which achieve higher (30%+) mAP, require large GPU memory to train the model mainly due to the large batch and network sizes. For example, RetinaNet uses 8 GPUs with a minibatch size of 16 (2 images per GPU). Therefore, with our resources (4 1080Ti GPUs), it was challenging to train the SOTA detector algorithms under the same settings such as the batch and model sizes.

Last, there is a well known limitation of Semi-Supervised Learning that the performance of any Semi-Supervised Learning method usually degrades with out-of-class distribution unlabeled samples. In the case of our algorithms, when training the labeled VOC dataset with the unlabeled MSCOCO dataset, it shows

the performance degradation. This is the same trend as the classification problem, and in-class distribution data should be added also as unlabeled data is added.

6.3 Future Directions

To conclude, we discuss the future directions of the research in Semi-Supervised Learning for object detection and Interpolation Regularization.

First, a method of matching the outputs in a two-stage detector is needed. By matching outputs, we can expect performance improvement through training of consistency loss in RPN and ISD training in the two-stage detector. However, as mentioned in Limitations, it is difficult to match between two outputs due to the sampling.

Second, a method for memory efficiency is required. The proposed algorithm requires 2-3 times more memory than the conventional algorithm. Since the state-of-the-art algorithms with high performance recently allocate much GPU memory to train, it is difficult to additionally train Semi-Supervised Learning. Therefore, by efficiently using memory and applying Semi-Supervised Learning to the SOTA object detector, we can expect further performance improvement.

Third, processing for out-of-class distribution is essential. Since the problem of out-of-class distribution has not yet been solved, the classification of unlabeled data is vital. To address the out-of-class distribution, we can sample the unlabeled data using the trained model or ignore it during the training. If we do sampling, training can be efficient but there is a possibility of the in-class distribution not being added. On the other hand, if we ignore it in the training

phase, it can be inefficient to predict for all samples. Thus, a way to deal with the out-of-class distribution to allow training without human intervention must be proposed.

Fourth, experiments with different proportions of labeled and unlabeled data are required. Most of our experiments were performed at a fixed ratio. However, in recent studies, experiments are being conducted with changing proportions of labeled data (0.5%, 1%, 2%, 5%, and 10%). To control the ratio of labeled data, sampling is performed, and it makes a large deviation. Therefore, through 3 ~ 5 experiments, results are obtained, and they have reported the average and standard deviation.

Fifth, for research in this field, many GPUs are required, and a method to learn or utilize them efficiently is needed. When training our CSD+ISD experiment with the SSD300 model, the PASCAL VOC dataset takes about 1 day, and the MS COCO dataset takes about 3 days with four Nvidia 1080Ti GPUs. Therefore, our algorithm takes at least $3 \text{ days} \times 3 \text{ times} \times 5 \text{ ratio} = 45 \text{ days}$ for the MSCOCO experiment. It will take longer to learn the latest algorithm, and this should be taken into consideration.

Last, there are still many problems in Interpolation Regularization, and research on this is needed. We have shown good results with a very simple binary classification problem as an example in 2, but there are problems as in our ISD. An example of an IR problem similar to that in ISD can be found in the SVHN dataset that consists of digit numbers of data. When we mix 1 and 4, the mixed image would be similar to 4. At that time, it faces the same problem with ISD. We solve this problem to fit object detection, but for other problems, it is needed to solve the problem considering tasks.

Bibliography

- [1] C. Arth, F. Limberger, and H. Bischof. Real-time license plate recognition on an embedded dsp-platform. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- [2] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What’s the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pages 549–565. Springer, 2016.
- [3] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool. Pedestrian detection at 100 frames per second. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2903–2910. IEEE, 2012.
- [4] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations*, 2020.
- [5] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060, 2019.

- [6] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
- [7] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.
- [8] S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen. Automatic license plate recognition. *IEEE transactions on intelligent transportation systems*, 5(1):42–53, 2004.
- [9] O. Chapelle, B. Scholkopf, and A. Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.
- [10] A. Chawla, H. Yin, P. Molchanov, and J. Alvarez. Data-free knowledge distillation for object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3289–3298, 2021.
- [11] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker. Learning efficient object detection models with knowledge distillation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 742–751, 2017.
- [12] Z. Chen, Z. Fu, R. Jiang, Y. Chen, and X.-S. Hua. Slv: Spatial likelihood voting for weakly supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12995–13004, 2020.
- [13] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019.

- [14] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.
- [15] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.
- [16] T. Devries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *CoRR*, abs/1708.04552, 2017.
- [17] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1532–1545, 2014.
- [18] P. Dollár, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. 2010.
- [19] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. 2009.
- [20] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):743–761, 2011.
- [21] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):743–761, 2012.
- [22] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6569–6578, 2019.

- [23] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.
- [24] G. French, A. Oliver, and T. Salimans. Milking cowmask for semi-supervised image classification. *arXiv preprint arXiv:2003.12022*, 2020.
- [25] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.
- [26] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf. Survey of pedestrian detection for advanced driver assistance systems. *IEEE transactions on pattern analysis and machine intelligence*, 32(7):1239–1258, 2009.
- [27] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- [28] M. Hyun, J. Jeong, and N. Kwak. Class-imbalanced semi-supervised learning. *arXiv preprint arXiv:2002.06815*, 2020.
- [29] Z. Jie, Y. Wei, X. Jin, J. Feng, and W. Liu. Deep self-taught learning for weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1377–1385, 2017.
- [30] D. Kim, G. Lee, J. Jeong, and N. Kwak. Tell me what they’re holding: Weakly-supervised object detection with transferable knowledge from human-object interaction. *arXiv preprint arXiv:1911.08141*, 2019.

- [31] J.-H. Kim, W. Choo, and H. O. Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, pages 5275–5285. PMLR, 2020.
- [32] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.
- [33] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.
- [34] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.
- [35] M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010.
- [36] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Mallocci, A. Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, pages 1–26, 2020.
- [37] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.
- [38] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

- [39] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti. A robust real-time automatic license plate recognition based on the yolo detector. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2018.
- [40] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.
- [41] H. Law, Y. Teng, O. Russakovsky, and J. Deng. Cornernet-lite: Efficient keypoint based object detection. *arXiv preprint arXiv:1904.08900*, 2019.
- [42] S. Lawrence, C. L. Giles, and A. C. Tsoi. What size neural network gives optimal generalization? convergence properties of backpropagation. Technical report, 1998.
- [43] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Light-head r-cnn: In defense of two-stage object detector. *arXiv preprint arXiv:1711.07264*, 2017.
- [44] Z. Li and F. Zhou. Fssd: feature fusion single shot multibox detector. *arXiv preprint arXiv:1712.00960*, 2017.
- [45] C.-H. Lin, Y.-S. Lin, and W.-C. Liu. An efficient license plate recognition system using convolution neural networks. In *2018 IEEE International Conference on Applied System Invention (ICASI)*, pages 224–227. IEEE, 2018.
- [46] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017.

- [47] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.
- [48] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [49] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [50] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.
- [51] G. J. McLachlan. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association*, 70(350):365–369, 1975.
- [52] T. Miyato, S.-i. Maeda, S. Ishii, and M. Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018.
- [53] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011.

- [54] N.-V. Nguyen, C. Rigaud, and J.-C. Burie. Semi-supervised object detection with unlabeled data. *In international conference on computer vision theory and applications*, 2019.
- [55] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246, 2018.
- [56] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.
- [57] J. Redmon, S. Divvala, R. Girshick, and A. 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.
- [58] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.
- [59] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
- [60] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- [61] Z. Ren, Z. Yu, X. Yang, M.-Y. Liu, Y. J. Lee, A. G. Schwing, and J. Kautz. Instance-aware, context-focused, and memory-efficient weakly supervised object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10598–10607, 2020.

- [62] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. 2005.
- [63] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [64] O. Russakovsky, L.-J. Li, and L. Fei-Fei. Best of both worlds: human-machine collaboration for object annotation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2121–2131, 2015.
- [65] M. Shi, H. Caesar, and V. Ferrari. Weakly supervised object localization using things and stuff transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3381–3390, 2017.
- [66] B. Singh, H. Li, A. Sharma, and L. S. Davis. R-fcn-3000 at 30fps: Decoupling detection and classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1081–1090, 2018.
- [67] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.
- [68] K. Sohn, Z. Zhang, C.-L. Li, H. Zhang, C.-Y. Lee, and T. Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020.
- [69] M. Tan, R. Pang, and Q. V. Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision*

- and pattern recognition*, pages 10781–10790, 2020.
- [70] Y. Tang, J. Wang, B. Gao, E. Dellandréa, R. Gaizauskas, and L. Chen. Large scale semi-supervised object detection using visual and semantic knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2119–2128, 2016.
- [71] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.
- [72] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9627–9636, 2019.
- [73] Y. Tokozume, Y. Ushiku, and T. Harada. Between-class learning for image classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [74] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.
- [75] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio. Manifold mixup: Better representations by interpolating hidden states. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6438–6447, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

- [76] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3635–3641. International Joint Conferences on Artificial Intelligence Organization, 7 2019.
- [77] V. Verma, M. Qu, K. Kawaguchi, A. Lamb, Y. Bengio, J. Kannala, and J. Tang. Graphmix: Improved training of gnns for semi-supervised learning. *arXiv e-prints*, pages arXiv–1909, 2019.
- [78] V. Verma, M. Qu, A. Lamb, Y. Bengio, J. Kannala, and J. Tang. Graphmix: Regularized training of graph neural networks for semi-supervised learning. *ArXiv*, abs/1909.11715, 2019.
- [79] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.
- [80] J. Wang, J. Yao, Y. Zhang, and R. Zhang. Collaborative learning for weakly supervised object detection. *arXiv preprint arXiv:1802.03531*, 2018.
- [81] K. Wang, L. Lin, X. Yan, Z. Chen, D. Zhang, and L. Zhang. Cost-effective object detection: Active sample mining with switchable selection criteria. *IEEE Transactions on Neural Networks and Learning Systems*, (99):1–17, 2018.
- [82] K. Wang, X. Yan, D. Zhang, L. Zhang, and L. 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.

- [83] R. J. Wang, X. Li, and C. X. Ling. Pelee: A real-time object detection system on mobile devices. *arXiv preprint arXiv:1804.06882*, 2018.
- [84] C. Wei, K. Sohn, C. Mellina, A. Yuille, and F. Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. *arXiv preprint arXiv:2102.09559*, 2021.
- [85] S. Woo, S. Hwang, and I. S. Kweon. Stairnet: Top-down semantic aggregation for accurate one shot detection. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1093–1102. IEEE, 2018.
- [86] Q. Xie, M.-T. Luong, E. Hovy, and Q. 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.
- [87] Z. Yan, J. Liang, W. Pan, J. Li, and C. Zhang. Weakly-and semi-supervised object detection with expectation-maximization algorithm. *arXiv preprint arXiv:1702.08740*, 2017.
- [88] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *arXiv preprint arXiv:1905.04899*, 2019.
- [89] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.
- [90] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. Single-shot refinement neural network for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4203–4212, 2018.

- [91] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1491–1498. IEEE, 2006.
- [92] X. Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2(3):4, 2006.
- [93] X. Zhu. Semi-supervised learning tutorial. In *International Conference on Machine Learning (ICML)*, pages 1–135, 2007.
- [94] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919, 2003.
- [95] X. Zhu and A. B. Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.
- [96] Y. Zhu, Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao. Soft proposal networks for weakly supervised object localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1841–1850, 2017.
- [97] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014.

초 록

객체 검출 알고리즘은 RGB 이미지에서 어느 위치에 어떤 객체가 있는지를 검출하는 것으로, 컴퓨터 비전 분야에서 가장 중요한 연구분야 중 하나이다. 하지만, 이러한 객체 검출 알고리즘을 위해서는 잘 레이블링된 큰 데이터 셋을 필요로 하고, 이러한 레이블링은 매우 많은 비용과 시간을 필요로 한다. 위와 같은 문제를 해결하기 위하여 약한 지도학습 (Weakly Supervised Learning), 준지도 학습 (Semi Supervised Learning)의 방법들이 연구되고 있으나, 그 연구가 많지 않고, 준지도 학습의 경우, 최신 딥러닝 기반의 학습방법들이 적용되지 않고 있었다.

본 논문에서는 최신의 딥러닝 기반의 준지도 학습 방법들을 객체 검출 알고리즘에 적용하였고, 여기서 발생하는 문제들을 발견하고 해결하는 방법을 연구하였다. 구체적으로 일관성 정규화 (Consistency Regularization), 보간법 정규화 (Interpolation Regularization) 기반의 준지도 학습 방법을 제시하였고, 최종적으로 이 둘을 합치는 방법을 제시하였다. 이는 기존의 분류문제에서 사용되는 CR 과 IR 을 객체검출 알고리즘 문제에 처음으로 확장한 것이다.

첫 번째로, 우리는 객체 검출 알고리즘을 위한 일관성 정규화 기반의 준지도 학습방법 (CSD)을 제안하였다. 이는 정규화 제약을 사용하여 레이블링이 없는 모든 데이터를 활용하여 객체 검출 성능을 향상시키는 방법이다. 구체적으로 우리는 정규화 제약을 분류뿐만 아니라 회귀에 대해서도 적용하였다.

게다가, 우리는 한 이미지 내에서 대부분의 영역을 차지하는 배경 부분의 영향을 줄이기 위하여 배경 제거 (Background Elimination) 을 적용하였다. 우리는 제안한 CSD 를 싱글 단계 (Single-Stage)와 두 단계(Two-Stage) 검출기에 모두 적용하여 평가하였고, 결과들은 우리의 알고리즘의 효과를 보였다.

두번째로, 우리는 객체 검출 알고리즘을 위한 보간법 정규화 (IR) 기반의 준지도 학습방법 (ISD)을 제안하였다. 우리는 보간법 정규화를 객체 검출 알고리즘에 바로 적용시켰을 때 생기는 문제들을 고려하고 해결하였다. 우리는 두 원본 패치에서의 객체 확률에 따라 모델의 출력을 두개의 타입으로 나누었다. 그리고, 우리는 각각의 타입에 따라 각각에 맞는 손실 함수를 정의하였다. 제안한 알고리즘은 지도학습뿐만 아니라 준지도 학습에서도 매우 큰 성능향상을 보였다.

마지막으로, 우리는 위의 CSD 와 ISD 의 결합하는 방법을 소개하였다. CSD에서는 일관성 정규화를 적용하기 위하여 한번의 추가적인 연산을 필요로 하고, 이는 기존의 지도학습에 비해 2배의 메모리를 필요로 한다. ISD의 경우, 보간법 정규화를 적용하기 위하여 두번의 추가적인 연산을 필요로 하고, 이는 3배의 메모리를 필요로 한다. 그러므로, 두 알고리즘을 결합하기 위해서는 세번의 추가적인 결과값이 필요하다. 우리는 CSD 미니배치의 샘플들을 섞는 방법을 적용하였고, 이는 추가적인 연산을 세번에서 두번으로 줄여 메모리의 소모를 줄일 수 있었다. 또한, 이 두 알고리즘을 합쳐서 모델의 성능이 향상됨을 보였다.

주요어: 준지도학습, 객체검출 알고리즘, 일관성 정규화, 보간법 정규화, 딥러닝
학번: 2015-26109

감사의 글

제 석박통합 기간 동안 많은 도움을 주신 분들께 진심으로 감사드립니다. 학문적, 인격적으로 제가 존경하는 저의 지도 교수님, 연구에 많은 도움을 주었던 공동 저자분들, 과제 수행으로 같이 고생하던 동료들, 같이 수학한 연구실 분들, 저를 지지해 주고 응원해 준 가족 친구분들께 모두 감사드립니다.

우선, 존경하는 지도교수 곽노준 교수님께 감사드립니다. 신입생 때부터 저의 고민, 아이디어 등에 대해서 누구보다 진지하게 같이 고민해 주셔서 감사드립니다. 학생들이 과제 등으로 문제를 겪을 때, 먼저 나서서 학생들이 학업에 집중할 수 있도록 도와주신 덕분에 마음껏 연구할 수 있었습니다. 또한, 사소한 아이디어일지라도 다양한 의견을 내주시고, 논문 작성에도 누구보다 꼼꼼하게 체크해 주신 덕분에 작은 하나하나 많이 배울 수 있었습니다. 석박통합 기간 6년 동안 한 명의 연구자로서 성장하는데 많은 지원과 지도를 해주신 것에 진심으로 감사드립니다.

그리고 학부 시절 지도교수 김영진 교수님께도 감사드립니다. 노력은 배신하지 않는 법이라며 항상 노력하시는 교수님을 보며 많은 자극을 받았었습니다. 대학원 생활 중에도 간혹 찾아가면 항상 반겨주시고 좋은 말씀해 주신 교수님께 진심으로 감사드립니다.

다음으로, 학위논문 심사를 위해 수고해 주신 심사위원 이교구 교수님, 이원종 교수님, 서봉원 교수님, 이민식 교수님께 진심으로 감사드립니다.

그리고 석박통합 기간 동안 함께 지낸 연구실의 많은 분들께 감사합니다.

우선, 저와 같이 논문 작성하며 고생했던 많은 분들께 감사드립니다. BMVC 논문을 같이 작성한 효진이, NeurIPS 작성을 같이한 승의형, 김지수, AAAI 작성을 같이한 대식이형, 규정이형, CVPR 작성을 같이한 Vikas, 민성이형 모두에게 감사합니다. 특히, 승의형, 김지수, 민성이형에게는 제가 정말 많은 도움을 받았는데, 도움을 받은 만큼 여러분들을 많이 도와드리지 못한 것 같아 죄송스러운 마음을 가지고 있고 진심으로 감사합니다.

그리고, 저와 같이 과제를 수행하며 고생해 주신 분들께도 감사드립니다. 특히 4년짜리 과제를 같이 수행하며 고생한 경민이형, 재석이, 건석이에게 감사합니다. 큰 프로젝트이다 보니, 일도 많았고 힘든 시간도 많았는데, 여러분들과 같이해서 해낼 수 있었습니다. 감사합니다.

연구실 다른 분들에게도 감사드립니다. 신입생 때부터 많이 챙겨주신, 오지용박사님, 지은누나, 헤민누나, 혁진이형, 성현이형에게 먼저 감사를 드립니다. 정치, 사회, 장학, 연구, 스포츠 등 다양한 얘기들을 같이 한 209호 식구들, 지훈이형, 규정이형, 승의형, 한열이, 김지수, 호준이에게도 감사드립니다. 그리고, 비슷한 시기에 들어와서 오랜 시간 동안 같이 수학하고 연구 토론을 같이 한 지혜누나, 재영이, 상호, 장호, 재석이, 성욱이에게도 감사드립니다. 회사에서 오셔서 연구뿐만 아니라 다양한 회사 문화, 산학 프로그램 등 많은 조언을 해주셨던, 영규형님, 영수형님, 승의형, 선훈이형, 동석이형, 원민형님, 박샘형님, 시명이형님 모두 감사드립니다.

항상 응원해 주고 격려해 준 가족, 친구들께 진심으로 감사합니다. 육체적, 정신적 성장에 도움을 주신 부모님, 동생에게도 감사드립니다. 한 번씩 소주 한잔하며, 같이 늙어가는 나의 가장 친한 친구들 윤우, 선종이, 성운이에게도 감사드립니다. 마지막으로 저의 결정을 존중하여 본인의 커리어를 포기하고 타지행을 같이 하기로 해준 저의 아내에게도 진심으로 감사드립니다.

이제 오랜 학업 끝에 사회생활을 시작하게 되었습니다. 그럼에도 학생의 마음으로 항상 배우고 발전하도록 노력하겠습니다. 감사합니다.