Interpolation-based Semi-supervised Learning for Object Detection

Jisoo Jeong[1] Vikas Verma[2,3] Minsung Hyun[1,4] Juho Kannala[2] Nojun Kwak[1] *
[1]Seoul National University, Seoul, Korea, [2]Aalto University, Finland,
[3]Mila - Québec Artificial Intelligence Institute, Montréal, Canada, [4]SK hynix

Abstract

Despite the data labeling cost for the object detection tasks being substantially more than that of the classification tasks, semi-supervised learning methods for object detection have not been studied much. In this paper, we propose an 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. In the supervised learning setting, our method improves the baseline methods by a significant margin. In the semi-supervised learning setting, our algorithm improves the performance on a benchmark dataset (PASCAL VOC and MSCOCO) in a benchmark architecture (SSD). Our code is available at https://github.com/soo89/ISD-SSD

1. Introduction

A dataset for object detection is much harder to create than the one for classification. While there is only one class in a single image for the classification task, there are multiple objects with different class labels in a single image for the object detection task. Therefore, the dataset for supervised object detection requires a pair of a class label and bounding box information for each object. Labeling each object takes more than a few seconds, and creating these datasets requires hundreds of hours [19, 1, 9].

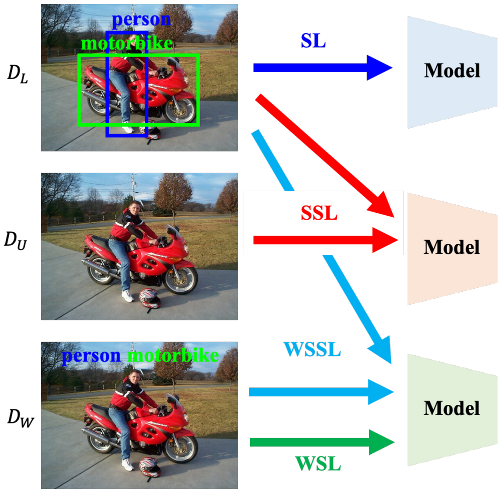

Figure 1. Supervised Learning (SL), Semi-supervised Learning (SSL), Weakly-Supervised Learning (WSL) and Weakly Semi-Supervised Learning (WSSL) for Object Detection. In this paper, we deal with SSL.

Due to the higher time and resource complexity for creating object detection datasets, recently, methods for learning with weakly labeled data (D_W)[1] or unlabeled data (D_U) have been studied as opposed to learning with the labeled data (D_L) only. There are mainly three types of object detection methods: weakly-supervised, semi-supervised, and weakly-semi-supervised learning. Weakly-supervised learning trains a model with a dataset that has only class information but no location information (D_W) [34, 20, 12, 28, 13]. On the other hand, weakly-semi-supervised learning is a learning method which uses D_W as well as D_L [22, 30]. Weakly-semi-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) [29, 18, 11] (See Fig. 1.).

In this paper, 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

*corresponding author

[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.

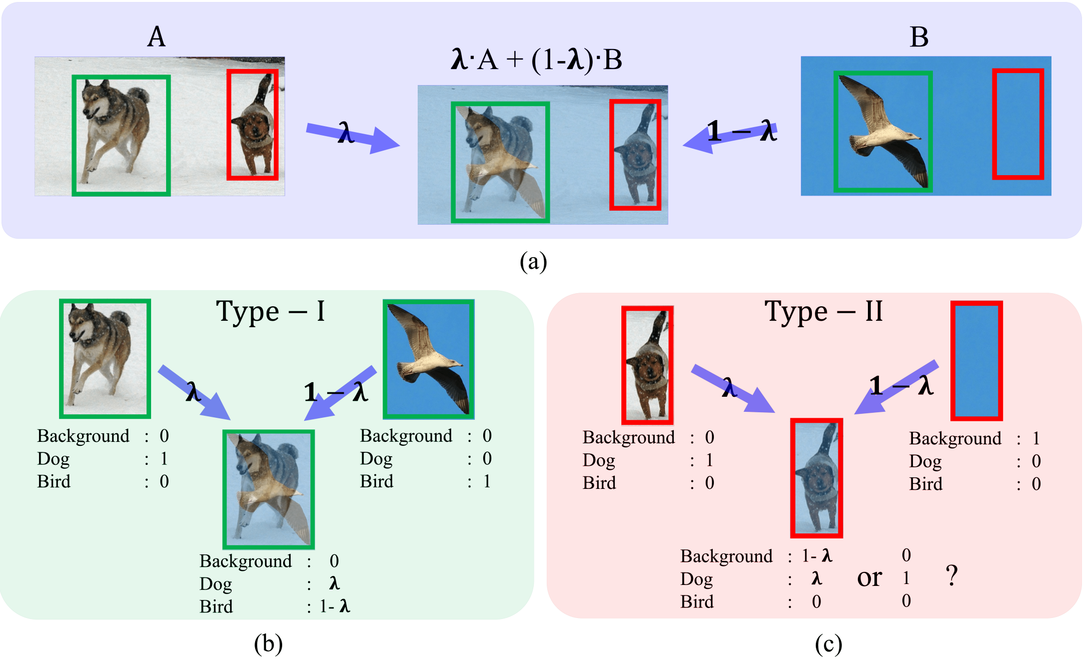

Figure 2. (a) Mixed image created by random interpolation between images A and B (b) Type-I : both patches are from object classes. (c) Type-II : one of the patches is from the background class.

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 [32, 25, 26, 4, 27]. However, it is challenging to apply IR directly to object detection because the background class consists of a very diverse and irregular texture. Fig. 2 shows an example of applying IR to the object detection problem. In Fig. 2(a), 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. 2(b). However, when an object is mixed with a background as in Fig. 2(c), 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 paper, 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 Sec. 3 can be combined with conventional semi-supervised learning methods such as CSD (consistency-based semi-supervised learning) [11] 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 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 algorithms achieving SOTA semi-supervised object detection performance.

2. Related Work

2.1. Interpolation-based Regularization (IR)

Interpolation-based Regularization is a promising approach due to its state-of-the-art performances and virtually no additional computational cost. These methods construct additional training samples by combining two or more training samples. Mixup [32] and Between-class learning [24] 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 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. Manifold Mixup [25] mixes features in the deeper layers instead of input images. Other works such as CutMix [31] construct the synthetic samples by mixing the CutOut [8] 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 unsupervised learning setting, interpolation-based regularizers have been explored in ACAI [5] and AMR [2]. These methods learn better unsupervised representations by enforcing a constraint that the representations obtained by mixing the representations of two samples should correspond to a data point on the data manifold.

2.2. Semi-Supervised Learning (SSL)

Semi-Supervised Learning (SSL) is a dominant approach for machine learning when the annotated data is scarce. There has been recent surge of interest in deep learning based on SSL for object classification [26, 4, 27]. These methods can be broadly categorized into: (1) consistency regularization methods and (2) generative adversarial networks (GAN) based methods. We describe below focusing on consistency regularization methods, which is highly relevant to out research.

The central idea of the consistency regularization methods is to enforce that the model predictions should not change under *reasonable* permutations to the input. For object classification, such permutations entail random translation, random cropping and horizontal flipping etc. Let us assume that x and x' are the original and the permuted 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:

$$L_U = d(f(x), f(x')) \tag{1}$$

$$L_{total} = L_S + w(t) \cdot L_U. \tag{2}$$

Some notable examples of consistency training include Π model [14], virtual adversarial training [17] and Mean Teacher [23]. The recent advances in this direction includes interpolation consistency training (ICT) [26] (its variants MixMatch [4], ReMixMatch [3]) and FixMatch [21].

ICT, which is a specific type of consistency regularization, constructs additional training samples through random interpolation of two different unlabeled images. The consistency loss of ICT consists of the output of interpolated image and the interpolated output of two images. In addition to this, ICT predicts their outputs with a teacher model (Mean-teacher [23]), which is an exponentially moving averaged network during training.

FixMatch uses another form of consistency regularization, where the model's prediction on "weak augmentation" are encouraged to be consistent with the "strong augmentation". For weak augmentation, FixMatch uses horizontal flipping, random translation and cropping, and for strong augmentation it uses Cutout [8], RandAugment [7] and CTAugment [3].

2.3. IR for Object Detection

Interpolation Regularization for Object Detection has recently been studied in [33, 6]. They applied IR to object detection in a supervised manner, and they focused on the distribution and the mixed object region. However, they did not consider the relationship between a foreground object and the background (Our Type-II). In this paper, different from the previous algorithms, we propose a method that applies IR to semi-supervised learning while considering the relationship between an object and the background.

2.4. SSL for Object Detection

Semi-Supervised Learning for Object Detection has recently been studied in [11] where CSD, the first consistency-regularization-based semi-supervised object detection method, was proposed. They exploited the consistency between the output predictions from the original image and the horizontally flipped one. Using the horizontal flip perturbation, it easily computes the consistency losses of classification and box regression at each position. To prevent the 'background' class from dominating the consistency loss in Eq. (2), they proposed the Background Elimination (BE) method which excludes boxes with high background probability in the computation of the consistency loss. In this paper, we also utilize the BE using the class probability of each candidate box. Also, the proposed ISD is combined with CSD to produce the SOTA SSL object detection performance.

3. Method

We denote a horizontally flipped version of an image A as $\hat{A}$, and the image created by random mixing, $\lambda \cdot A + (1-\lambda) \cdot B$, of two images A and B as $Mix_\lambda(A, B)$. Similar to Mixup, the mixing coefficient λ is drawn from the $Beta(\alpha, \alpha)$ distribution. In our method, we use SSD [16], one of the most popular single-stage object detectors, as a baseline detector. In the training of SSD, we add the newly proposed interpolation-based consistency regularization loss in combination with the flip-based consistency regularization loss in [11] to enhance the performance. The network output of SSD $f^{p,r,c,d}$ is denoted as the output of

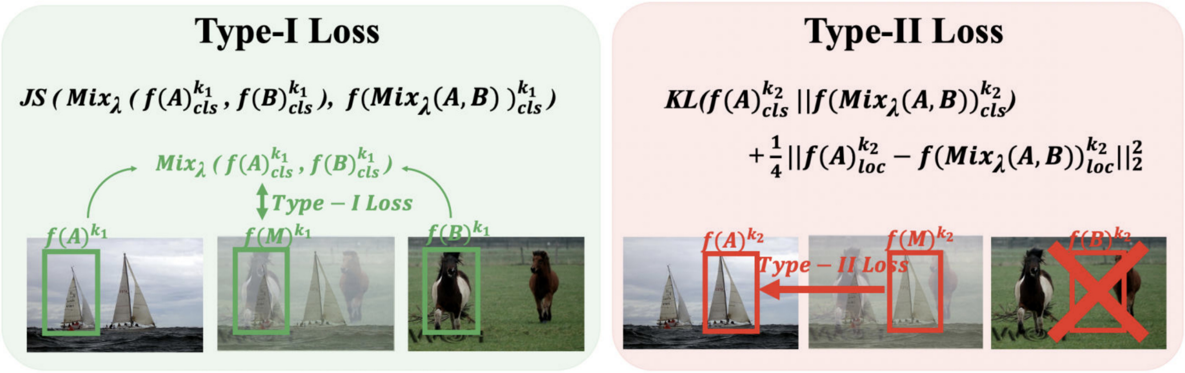
Type-I Loss
$JS\,(Mix_\lambda\,(f(A)^{k_1}_{cls}, f(B)^{k_1}_{cls}),\ f(Mix_\lambda(A,B))^{k_1}_{cls})$
$Mix_\lambda\,(f(A)^{k_1}_{cls}, f(B)^{k_1}_{cls})$
$Type-I\ Loss$
$f(A)^{k_1}$
$f(M)^{k_1}$
$f(B)^{k_1}$
Type-II Loss
$KL(f(A)^{k_2}_{cls}\,||f(Mix_\lambda(A,B))^{k_2}_{cls})$
$+\frac{1}{4}||f(A)^{k_2}_{loc} - f(Mix_\lambda(A,B))^{k_2}_{loc}||^2_2$
$f(A)^{k_2}$
$f(M)^{k_2}$
$f(B)^{k_2}$
$Type-II\ Loss$

Figure 3. The proposed ISD loss for each type. $Mix_\lambda(a,b) = \lambda \cdot a + (1-\lambda)\cdot b$

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^k_{cls} and f^k_{loc} 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:

$$m(I)^k = \begin{cases} 1, & \text{if } \operatorname{argmax}(f^k_{cls}(I)) \neq background \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

3.1. Interpolation-based Semi-supervised learning for Object Detection (ISD)

3.1.1 Type categorization.

We determine the type of a pair of patches by the background elimination method [11] that only extracts patches with a high objectness probability. Then we apply different methods appropriate for each type of patches. Eq. (4) 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^A_{II} 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^B_{II}.

$$\begin{aligned} \text{Type-I mask:} \quad & m_I = m(A) \otimes m(B), \\ \text{Type-II(A) mask:} \quad & m^A_{II} = m(A) \otimes \sim m(B), \\ \text{Type-II(B) mask:} \quad & m^B_{II} = \sim m(A) \otimes m(B). \end{aligned} \tag{4}$$

3.1.2 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 [26]. Note that there are two differences compared to the conventional ICT. First, we used *α-Jensen-Shannon divergence* (JSD / for α=1) as the consistency regularization loss (function $d(.,.)$ in Eq. (2)). In the CSD, JSD 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 [23]. Eq. (5) shows the loss function of Type-I, which is the distance between the mixed output of $f(A)^k_{cls}$ and $f(B)^k_{cls}$ and the output of the mixed image of A and B, $f(Mix_\lambda(A,B))^k_{cls}$.

$$l_I = JS(Mix_\lambda(f(A)^k_{cls}, f(B)^k_{cls})||f(Mix_\lambda(A,B))^k_{cls}) \tag{5}$$

The overall Type-I loss $\mathcal{L}_I$ is the average of patches whose Type-I mask is 1, i.e. $\mathcal{L}_I = \mathbb{E}_{\mathbb{I}\{m_I=1\}}[l_I]$. Here, $\mathbb{E}$ and $\mathbb{I}$ are the expectation and the indicator function, respectively.

3.1.3 Type II loss

As shown in Fig. 3, 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 [21] which encourages consistency between the predictions on the strong augmentation 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

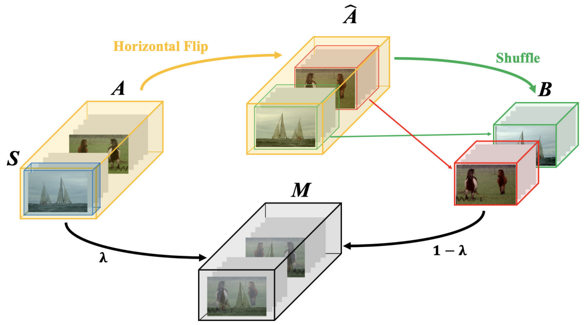

Figure 4. Combination of ISD with CSD. The original images ($\mathcal{A}$) are flipped ($\hat{\mathcal{A}}$) and the mixed images ($\mathcal{M}$) are obtained by combining the two. First, the order of flipped images are changed by shuffling ($\mathcal{B}$ = shuffle($\hat{\mathcal{A}}$)), then $\mathcal{A}$ and $\mathcal{B}$ are mixed ($\mathcal{M} = Mix_\lambda(\mathcal{A}, \mathcal{B})$). 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.

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 ($f(Mix_\lambda(A, B))$) 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:

$$l^A_{II\_cls} = KL(f(A)^k_{cls}||f(Mix_\lambda(A, B))^k_{cls}) \qquad (6)$$

$$l^A_{II\_loc} = \frac{1}{4}\|f(A)^k_{loc} - f(Mix_\lambda(A, B))^k_{loc}\|^2_2. \qquad (7)$$

The overall Type-II loss when patch A is foreground, $\mathcal{L}^A_{II}$, is calculated as the average of the sum of two individual losses as $\mathcal{L}^A_{II} = \mathbb{E}_{\mathbb{I}\{m^A_{II}=1\}}[l^A_{II\_cls} + l^A_{II\_loc}]$. Likewise, $\mathcal{L}^B_{II}$ is also calculated by applying the above loss, and the total loss of Type-II is calculated as $\mathcal{L}_{II} = \mathcal{L}^A_{II} + \mathcal{L}^B_{II}$.

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

$$\mathcal{L}_{ISD} = \gamma_1 \cdot \mathcal{L}_I + \gamma_2 \cdot \mathcal{L}_{II}. \qquad (8)$$

Here, γ_1 and γ_2 are set appropriately to balance both loss terms.

Algorithm 1 Training procedure of the proposed ISD

```
Require: $\mathcal{D_L}, \mathcal{D_U}$: 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
for each $t \in [1, \text{max\_iterations}]$ do
    Data Preparation
        $\mathcal{A} \leftarrow \mathcal{D_L} \cup \mathcal{D_U},\ \hat{\mathcal{A}} \leftarrow h(\mathcal{A})$
        $\mathcal{B} \leftarrow shuffle(\hat{\mathcal{A}})$
        $\mathcal{C} \leftarrow Mix_\lambda(\mathcal{A}, \mathcal{B})$
    Compute the outputs
        $f(\mathcal{A}), f(\hat{\mathcal{A}}), f(\mathcal{C})$
        $f(\mathcal{B}) \leftarrow shuffle(f(\hat{\mathcal{A}}))$
    Compute the objectness mask
        $m_\mathcal{A} \leftarrow f(\mathcal{A}), \quad m_\mathcal{B} \leftarrow f(\mathcal{B})$          $(Eq.\ 3)$
    Compute the supervised & CSD losses
        $\mathcal{L_S} \leftarrow f(A \in \mathcal{D_L} \cap \mathcal{A})$
        $\mathcal{L}_{CSD} \leftarrow f(A \in \mathcal{D_U} \cap \mathcal{A}), f(\hat{A}), m_\mathcal{A}$
    Compute the ISD loss using the type mask (Eq. 4)
        $\mathcal{L}_I \leftarrow \mathbb{E}_{\mathbb{I}\{m_I=1\}}[l_I]$          $(Eq.\ 5)$
        $\mathcal{L}^A_{II} \leftarrow \mathbb{E}_{\mathbb{I}\{m^A_{II}=1\}}[l^A_{II\_cls} + l^A_{II\_loc}]$   $(Eq.\ 6, 7)$
        $\mathcal{L}^B_{II} \leftarrow \mathbb{E}_{\mathbb{I}\{m^B_{II}=1\}}[l^B_{II\_cls} + l^B_{II\_loc}]$
        $\mathcal{L}_{II} \leftarrow \mathcal{L}^A_{II} + \mathcal{L}^B_{II}$
        $\mathcal{L}_{ISD} \leftarrow \lambda_1 \cdot \mathcal{L}_I + \lambda_2 \cdot \mathcal{L}_{II}$
    Compute the total loss
        $\mathcal{L}_{Total} \leftarrow \mathcal{L_S} + w(t) \cdot (\mathcal{L}_{CSD} + \mathcal{L}_{ISD})$
    Update $f(\cdot)$ using $\mathcal{L}_{Total}$
end for
```

3.2. Combination of ISD with CSD

For ISD training, three sets of image batches, $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{M} = Mix_\lambda(\mathcal{A}, \mathcal{B})$ should be inferred by the network. For efficient training, we set $\mathcal{B}$ as the horizontally flipped version of $\mathcal{A}$, i.e, $\hat{\mathcal{A}} = flip(\mathcal{A})$, as shown in Fig. 4. We calculated the CSD loss with those two batches. However, 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. Therefore, as shown in Fig. 4, we make the mixed images by combining the original batch ($\mathcal{A}$) with the half-shuffled flipped batch ($\mathcal{B} = shuffle(\hat{\mathcal{A}})$). The total loss consists of supervised loss ($\mathcal{L_S}$), CSD loss ($\mathcal{L}_{CSD}$), and ISD loss ($\mathcal{L}_{ISD}$) as follows:

$$\mathcal{L}_{Total} = \mathcal{L_S} + w(t) \cdot [\mathcal{L}_{CSD} + \mathcal{L}_{ISD}], \qquad (9)$$

where $w(t)$ is a weight scheduling function. The overall process of the proposed semi-supervised learning is described in Algorithm 1

4. Experiments

4.1. Experimental Settings

Our experiments are based on pytorch. We have used a third-party code for SSD[2] and an official code for CSD[3]. We experimented on the PASCAL VOC dataset and MS COCO

[2] https://github.com/amdegroot/ssd.pytorch

[3] https://github.com/soo89/CSD-SSD

Table 1. Detection results for PASCAL VOC2007 test set under the supervised and the semi-supervised training setting. L_{cls} and L_{loc} are the consistency classification and localization loss with BE (Eq. 3) 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 (Supervised Learning) [16, 11]	VOC07	-	70.2
	VOC07 + VOC12	-	77.2
CSD [11]	VOC07	-	69.3
Ours (ISD only)		-	72.3
Ours (ISD + CSD)		-	73.1
Semi-Supervised Learning			
CSD [11] (L_{cls})	VOC07	VOC12	71.7 (1.5)
CSD [11] (L_{loc})			71.9 (1.7)
CSD [11] (L_{cls} + L_{loc})			72.3 (2.1)
Ours (ISD (Type-I only))	VOC07	VOC12	71.9 (1.7)
Ours (ISD (Type-II only))			73.8 (3.6)
Ours (ISD (Type-I,II))			74.1 (3.9)
Ours (CSD + ISD (Type-I,II))			74.4 (4.2)

dataset with SSD300 model. VGG-16 pre-trained model is used as our backbone network. PASCAL VOC [10] and MS COCO [15] data consist of 20 and 80 classes, respectively. For VOC dataset, we followed the settings from the conventional semi-supervised learning methods for object detection. Similar to [29, 11], 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 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.

We sample the mixing parameter λ from $Beta(\alpha, \alpha)$ at every iteration. The parameters are set to (γ_1, γ_2) = (0.1, 1) in Eq. (8) and α = 100 in the beta distribution. Other learning parameters such as the learning rate and the batch size are the same as [11].

4.2. PASCAL VOC

4.2.1 Supervised Learning

We start by examining the effect of ISD on SSD in the supervised training setting, i.e, the proposed losses in 9 are applied to labeled data. The results are presented in Table 1. In the first row block, SSD (base) trained with VOC 07 (*trainval*) data shows 70.2 mAP performance, while that of SSD (CSD) decreases to 69.3 mAP, which shows a clear side effect of over-regularization [4]. On the other hand, SSD300 (ISD) and SSD (ISD + CSD) 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.

4.2.2 Semi-Supervised Learning

We evaluate the performance of ISD in the SSL setting. As shown in Table 1, 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. In addition, 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 1, last row) shows 1.3% performance improvement in comparison to the fully supervised setting with VOC07 labeled data on SSD (Tabel 1, row 7). 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 1 demonstrate that our ISD+CSD approach outperforms the baseline CSD-only approach by a significant margin.

[4] We reported the score for the supervised CSD [11] in the their supplementary material

Table 2. 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
SSD [16]	Val35k	-	18.8	34.8	18.6
	Val35k + Train80k (trainval35k)	-	23.9	40.8	24.7
CSD [11]	Val 35k	Train 80k	19.8 (1.0)	35.8 (1.0)	19.8 (1.2)
Ours (CSD + ISD)			21.0 (2.2)	37.7 (2.9)	21.1 (2.5)

Table 3. 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

Table 4. 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**

4.3. MSCOCO

Table 2 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. Our proposed algorithm (CSD+ISD) shows 2.1% performance improvement in the same experimental setting for COCO dataset.

5. Discussion

5.1. 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 1. Each loss shows a significant performance improvement compared to the supervised learning. Furthermore, we report the combination of CSD and the different types of ISD losses in Table 3. 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, 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.

5.2. Beta distribution

In ISD, the mixing coefficient λ is sampled from the $Beta(\alpha, \alpha)$ distribution. Table 3 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, 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.

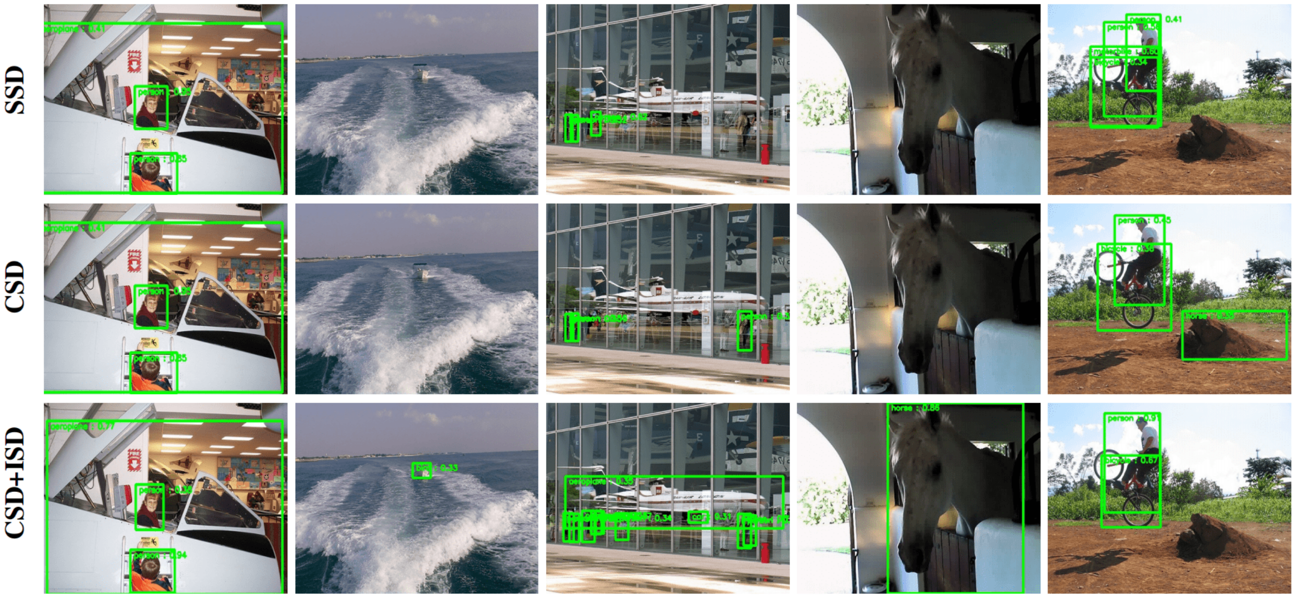

Figure 5. Qualitative results for the PASCAL VOC2007 test set using supervised SSD, semi-supervised CSD and CSD+ISD models in table 1. The first, middle, and last rows are the resulting images of the SSD, CSD, and CSD+ISD models, respectively. A score threshold of 0.3 is used to display these images. The images from the second column to the fourth column are the result when the image of the object is similar to the background or there is distortion. Our proposed algorithm shows that it works robustly in this situation. The results of the last column show that ISD does not detect all samples that look like objects.

5.3. 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.

5.4. Object detector

In this paper, 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.

6. Conclusion

In this paper, 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 different 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 further demonstrate that combining ISD with the previous method of CSD can further improve the performance. We leave the exploration of Interpolation Regularization for Two-stage detectors as a future work.

7. Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2021R1A2C3006659).

References

[1] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pages 549–565. Springer, 2016.

[2] Christopher Beckham, Sina Honari, Vikas Verma, Alex M Lamb, Farnoosh Ghadiri, R Devon Hjelm, Yoshua Bengio, and Chris Pal. On adversarial mixup resynthesis. In *Advances in Neural Information Processing Systems*, pages 4348–4359, 2019.

[3] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations*, 2020.

[4] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060, 2019.

[5] David Berthelot, Colin Raffel, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. In *International Conference on Learning Representations*, 2019.

[6] Shahine Bouabid and Vincent Delaitre. Mixup regularization for region proposal based object detectors. *arXiv preprint arXiv:2003.02065*, 2020.

[7] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019.

[8] Terrance Devries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *CoRR*, abs/1708.04552, 2017.

[9] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):743–761, 2012.

[10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[11] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. In *Advances in Neural Information Processing Systems*, pages 10758–10767, 2019.

[12] Zequn Jie, Yunchao Wei, Xiaojie Jin, Jiashi Feng, and Wei 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.

[13] Daesik Kim, Gyujeong Lee, Jisoo Jeong, and Nojun Kwak. Tell me what they're holding: Weakly-supervised object detection with transferable knowledge from human-object interaction. *arXiv preprint arXiv:1911.08141*, 2019.

[14] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

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

[17] Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[18] Nhu-Van Nguyen, Christophe Rigaud, and Jean-Christophe Burie. Semi-supervised object detection with unlabeled data. *In international conference on computer vision theory and applications*, 2019.

[19] Olga Russakovsky, Li-Jia Li, and Li 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.

[20] Miaojing Shi, Holger Caesar, and Vittorio Ferrari. Weakly supervised object localization using things and stuff transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3381–3390, 2017.

[21] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

[22] Yuxing Tang, Josiah Wang, Boyang Gao, Emmanuel Dellandréa, Robert Gaizauskas, and Liming 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.

[23] Antti Tarvainen and Harri 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.

[24] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-class learning for image classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[25] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In Kamalika Chaudhuri and Ruslan 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.

[26] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelli-*

gence, IJCAI-19, pages 3635–3641. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[27] Vikas Verma, Meng Qu, Alex Lamb, Yoshua Bengio, Juho Kannala, and Jian Tang. Graphmix: Regularized training of graph neural networks for semi-supervised learning. *ArXiv*, abs/1909.11715, 2019.

[28] Jiajie Wang, Jiangchao Yao, Ya Zhang, and Rui Zhang. Collaborative learning for weakly supervised object detection. *arXiv preprint arXiv:1802.03531*, 2018.

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

[30] Ziang Yan, Jian Liang, Weishen Pan, Jin Li, and Changshui Zhang. Weakly-and semi-supervised object detection with expectation-maximization algorithm. *arXiv preprint arXiv:1702.08740*, 2017.

[31] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *arXiv preprint arXiv:1905.04899*, 2019.

[32] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[33] Zhi Zhang, Tong He, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of freebies for training object detection neural networks. *arXiv preprint arXiv:1902.04103*, 2019.

[34] Yi Zhu, Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Soft proposal networks for weakly supervised object localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1841–1850, 2017.