Domain Adaptation using Self-Training with Mixup for One-Stage Object Detection

Jitender Maurya, Keyur R. Ranipa
Toshiba Software India Pvt. Ltd.
{jitender.maurya, keyur.ranipa}@toshiba-tsip.com

Osamu Yamaguchi, Tomoyuki Shibata, Daisuke Kobayashi
Corporate Research and Development Center, Toshiba Corporation
{osamu1.yamaguchi, tomoyuk11.shibata, daisuke32.kobayashi}@toshiba.co.jp

Abstract

In this paper, we present an end-to-end domain adaptation technique that utilizes both feature distribution alignment and Self-Training effectively for object detection. One set of methods for domain adaptation relies on feature distribution alignment and adapts models on an unlabeled target domain by learning domain invariant representations through adversarial loss. Although this approach is effective, it may not be adequate or even have an adverse effect when domain shifts are large and inconsistent. Another set of methods utilizes Self-Training which relies on pseudo labels to approximate the target domain distribution directly. However, it can also have a negative impact on the model performance due to erroneous pseudo labels. To overcome these two issues, we propose to generate reliable pseudo labels through feature distribution alignment and data distillation. Further, to minimize the adverse effect of incorrect pseudo labels during Self-Training we employ interpolation-based consistency regularization called mixup. While distribution alignment helps in generating more accurate pseudo labels, mixup regularization of Self-Training reduces the adverse effect of less accurate pseudo labels. Both approaches supplement each other and achieve effective adaptation on the target domain which we demonstrate through extensive experiments on one-stage object detector. Experiment results show that our approach achieves a significant performance improvement on multiple benchmark datasets.

1. Introduction

Object detection is a significant task in computer vision. With recent advances in deep learning and the availability of large-scale datasets, the deep Convolutional Neural Networks (CNNs) based object detectors [17, 8, 21] have significantly improved the performance of the object detection task on benchmark datasets [6, 16]. However, it is challenging to directly utilize an object detector trained on these benchmark datasets (source dataset) in an application where the dataset (target dataset) has a significant domain gap. This problem can be addressed with augmenting the source dataset with annotated target dataset. However, obtaining object annotations for each new domain dataset is time consuming, costly, and requires significant human efforts.

Unsupervised Domain Adaptation (UDA) is an approach to address the issue of domain shifts between label-rich source domain dataset and label-scarce target domain. Various methods for UDA have been proposed, of which most
aim to reduce domain shifts by aligning source and target domain feature distributions [7, 19, 26]. While these methods focus on the classification task, UDA methods for object detection have recently gained much attention [4, 25, 14, 37, 27, 33, 2, 9, 32, 36]. These state-of-the-art UDA methods for domain adaptive object detection attempt to align feature distribution of source and target domain by introducing adversarial loss. In this paper, training a model with an adversarial loss is denoted as Feature Distribution Alignment (FDA) step. Although these methods achieve a considerable performance improvement in the target domain over baseline object detectors, it remains challenging to further boost the performance when the domain gap is large. Especially, in the case of object detection when instance level, intra-domain variations are higher, feature distribution alignment can be ineffective or might even exhibit an adverse performance.

Another approach to domain adaptive object detection is Self-Training [23, 14, 22]. Self-Training utilizes a model trained on label-rich source domain dataset to generate pseudo labels for target domain. These labels are then used to train a model on target domain to approximate real data distribution. However, traditional approaches of Self-Training can have a negative effect on the model performance for multiple reasons. First, due to domain gap, generated pseudo labels are erroneous. Second, these incorrect pseudo labels can reinforce themselves during the subsequent iterations of Self-Training and further amplify the error, resulting in the negative performance. Finally, these labels are generated from model’s own predictions and may not have any meaningful information for improving performance on the target domain. These issues lead to rapid degradation in the performance of the model when Self-Training is applied for a longer period.

In the proposed method we address these two issues. First, as shown in Figures 1 and 2, we utilize FDA as a supplement step which helps to generate reliable pseudo labels for the Self-Training. Due to FDA, model learns domain invariant feature representations which helps in generating more accurate pseudo labels of target domain samples. In FDA, we apply feature distribution alignment at multiple feature levels of an object detector. Then we use model trained with FDA to generate pseudo labels that aid the Self-Training on the target domain. To further improve the accuracy of pseudo labels, we utilize data distillation. Data distillation by ensembling predictions of the same model on multiple transformations of the input is a common strategy to boost the performance of the model at a test time [20]. In the proposed Periodic Data Distillation (PDD), pseudo labels are generated by ensembling the predictions of the model on multiple transformations of the target domain images. These labels are periodically updated during the Self-Training as the model with FDA gets better at predictions.

The combination of FDA, that reduces the domain gap between the source and target domain and PDD, helps in generating more accurate pseudo labels.

Further, to address the issue of less accurate pseudo labels, we propose a consistency regularization of Self-Training with mixup (STM) [35, 30]. Mixup is a regularization technique using interpolation-based data augmentation which provides robustness against incorrect pseudo labels. We increase the strength of mixup interpolation to generate samples further from the real training samples to make memorization of corrupt or random labels difficult for the model. In contrast, when pseudo labels are correct and consistent it is easier to learn from such samples. Further, it also enforces consistent predictions on samples which lie on decision boundary [31]. Therefore, in the proposed method we employ mixup regularization to enforce consistent predictions and reduce memorization of incorrect pseudo labels during Self-Training.

We demonstrate the effectiveness of the proposed method on one-stage object detector SSD (Single Shot MultiBox Detector) [17]. Simplicity and a higher frame rate make one-stage object detectors more suitable for real-world applications. Further, domain adaptation methods based on one-stage detectors are limited. Therefore, we apply the proposed method to one-stage object detector. The main contributions of this paper are summarized as follows:

- We propose a method to generate accurate pseudo labels through Feature Distribution Alignment and Periodic Data Distillation.

- We propose an end-to-end, Self-Training method with interpolation-based consistency regularization called mixup and confirm its validity in the context of domain adaptation.

- We validate the proposed method on a one-stage detector. We conduct extensive experiments and achieve a comparable performance to state-of-the-art methods on the standard domain adaptation benchmarking datasets such as Clipart1k, Watercolor2k, Comic2k and challenging real-world scenario of domain adaptation for the task of pedestrian detection.

2. Related Work

Domain Adaptation addresses the issue of domain shifts between the source and target domain. State-of-the-art domain adaptation techniques for object detectors can broadly be classified in three categories. First are domain transfer techniques where source domain images are first translated into target domain and then the detector is fine-tuned on both source and target domain dataset. Domain transfer techniques were utilized in [10, 22] to perform two-step
progressive domain adaptation on one-stage object detector SSD[17]. They adapt source images to target domain using style transfer [38], afterward features are aligned to reduce domain gap. The second set of methods, that aligns distribution of source and target domain using an adversarial loss, has been very effective in classification and object detection tasks [14, 7, 4, 25, 18, 26, 33]. For example, weak global alignment method proposed in [25] focuses on aligning features of hard-to-classify samples in target domain and Background Score Regularization (BSR) proposed in [14] uses adversarial loss to train a feature extractor to extract discriminative features of target domain. Implicit Instance-Invariant Network (I²Net) [3] is specifically tailored for the one-stage detector, which considers intra-class, intra-domain variations and object patterns to learn instance invariant features.

The third technique is based on Self-Training, which uses pseudo labels to guide training over unlabeled target-domain data. Self-Training has been used along with both, a domain transfer technique [22, 10, 34] and adversarial training technique [14]. In general, the approach in these methods is to reduce domain gap for the purpose of generating reliable pseudo labels and then using them to self-train the object detector on target domain dataset.

Mixup is an interpolation-based regularization method that has been used for both supervised [30, 35] and semi-supervised learning settings [31, 1]. Input mixup [35] performs an interpolation of input space, while manifold mixup [30] performs an interpolation of hidden state representation to improve robustness against adversarial samples in supervised learning settings. In semi-supervised learning of image classification task, it is used [31] to enforce consistent and low-entropy predictions [1] on interpolation of unlabeled samples. Further, for the task of object detection, [12] shows the effectiveness of interpolation-based regularization in semi-supervised learning settings, where primary objective is to improve performance of the detector on unlabeled samples. Instead of using unlabeled samples of the source domain dataset. In contrast, the proposed method focuses on improving the performance of the object detector on the target domain dataset which does not have any labeled samples.

The proposed method primarily employs two techniques for building a domain adaptive object detector, Feature Distribution Alignment (FDA) and Self-Training (ST). We utilize FDA to reduce domain shifts and thus generate more accurate pseudo labels of target domain samples. These pseudo labels are then used in Self-Training of object detector on the target domain. Self-Training can be highly unstable and can have detrimental effect on the performance of the model in presence of incorrect pseudo labels. Therefore, we apply mixup based consistency regularization to reduce the adverse effect of incorrect pseudo labels.

3. Proposed Method

In Unsupervised Domain Adaptation setting of object detectors, we are given a set of images \( X_s \) from the source domain with labels \( Y_s \) and unlabeled set of images \( X_t \) from the target domain. \((x_s, y_s)\) is a sample drawn from \( X_s \).
where $x_s$ is an image and $y_s = \{(c_i, l_i)\}_{i=1}^m$ are corresponding $m$ object labels, where $c_i$ is the class label and $l_i$ is the location of the $i$th object within image. Further, $x_t$ is a sample drawn from $X_t$ and corresponding object labels are not available. $P(X_s)$ and $P(X_t)$ denote source and target domain distributions respectively and $P(X_s) \neq P(X_t)$. The illustration of components of the end-to-end proposed method is shown in Figure 2.

### 3.1. Feature Distribution Alignment

The primary objective of the Feature Distribution Alignment is to reduce the large domain distribution gap between $P(X_s)$ and $P(X_t)$. To achieve this objective, various methods have been proposed [7, 19] that align feature distribution by minimizing the distance between two domains. In general, this is done through attaching domain classifiers to feature extractor at various feature levels [25]. Then, the feature extractor and domain classifiers are trained in an adversarial way. This approach has been applied for various vision applications such as image classification and object detection. Recently, Saito et al. [25] proposed a Strong and Weak domain adaptation for the two-stage detector Faster RCNN [21]. They apply a Weak alignment to global features to alleviate the negative effect of feature alignment when domain shifts are large. For Weak alignment, they utilize Focal Loss (FL) [15] instead of cross-entropy loss as an objective for the domain classifier of global level features. For local features, such as texture and color, at lower layers of feature extractor, they apply Strong alignment by employing least-squares loss as an objective for the domain classifier. In the proposed method, FDA step is based on this idea but for one-stage object detector Figure 2 (a). We employ Weak alignment at global level features of the feature extractor. However, for the Strong alignment of local level features, we employ two domain classifiers at two different lower layers of the feature extractor. This helps in achieving alignment features not just at local level, but also of features at intermediate level which might constitute object features. By reducing the domain shifts, FDA helps in generating more reliable pseudo labels for the target domain dataset as compared to the model that is trained only on the source domain dataset. This is the primary motivation behind adding this module in the proposed method. The adversarial loss of FDA step is summarized as follows:

\[
L_{FDA} = L_{global} + L_{loc1} + L_{loc2}
\]

where $L_{global}$ is same as proposed in [25]. As we have two local classifiers at different layers, losses corresponding to them are denoted as $L_{loc1}$ and $L_{loc2}$ and they are also as proposed in [25].

### 3.2. Pseudo Labels using Periodic Data Distillation

As FDA reduces a distribution gap, model gets better at predicting correct labels for the target domain images. However, a single prediction from the model may not always be accurate. Moreover, misclassified samples with high-confidence score may end up being pseudo labels. Such pseudo labels can have a negative impact on the performance as they have a capacity to amplify the error during Self-Training. Accuracy of pseudo labels is of utmost important for effective and stable Self-Training. Therefore, we propose a Periodic Data Distillation method where a multi-transform inference [20] is applied on the model to generate high-quality pseudo labels. Data distillation using multi-transform inference has been proven effective on variety of vision application [11, 13, 20] including object detection. In data distillation, a trained model is applied on the multiple transformations of unlabeled input image to get predictions. The predicted bounding boxes for each object from all the transformed images are ensembled to generate a pseudo label. This process will help to have correct predictions as it is more likely to have bounding boxes in a common region in most predictions.

Given a state of the model $F^T_{\theta}$ at epoch $T-1$ with model parameters $\theta$, and unlabeled target image sample $x_t$ with its set of $k$ geometric transformations $\{x_t^1, x_t^2, \ldots, x_t^k\}$, where superscript 1 represents original image without any transformations, then the pseudo label is produced as follows:

\[
y_t = \text{ensemble}\{F^{T-1}_\theta(x_t^1), F^{T-1}_\theta(x_t^2)\ldots F^{T-1}_\theta(x_t^k)\}
\]

Here, $y_t = \{(c_i, l_i)\}_{i=1}^m$ will be used as a pseudo label for $x_t$ during Self-Training of the model state $F^T_{\theta}$ at epoch $T$.

To generate pseudo labels, we first apply multi-transform inference on the image $x_t$. For each transformed image there are $N_c$ object predictions of class $c$. If the location of the object is represented by bounding box coordinates, then $B_c = \{B_{i,c}\}_{i=1}^{N_c}$ is a set of all such bounding boxes in the image. In the ensemble operation, we merge $k$ predictions $\hat{B}_c = \bigcup_{j=1}^k B_{j,c} = \{B_{j,c}\}_{j=1}^{N_c}$ of multi-transform inference. Because of the multi-transform inference, the generated boxes will be highly concentrated around the actual objects of interest. We exploit this aspect in the ensemble operation and then, the final bounding box coordinates $B'_{i,c}$ is an average

\[
B'_{i,c} = \frac{B_{i,c} + \sum_{N(B_{i,c})} B_{j,c}}{1 + \text{count}(N(B_{i,c}))}
\]

coordinates of each bounding box $B_{j,c} \in N(B_{i,c})$ (where $N(B_{i,c})$ denotes the set of boxes in $\hat{B}_c$ that overlap with $B_{i,c}$ by more than 0.5 on IoU metric).
Figure 3. A visualization on Clipart1k feature representations (a) Top: Feature representations of Person (purple-colored dots) and Chair class (olive-colored dots) learned by Baseline model at conv4_3 and conv7 layer. Bottom: Feature representation learned by FDA+STM for same classes. It can be observed that Self-Training has improved discriminative capabilities of the model on target domain. (b) Singular Value Decomposition of class specific representations at conv10_2 layer. With FDA+STM, it can be observed that it reduces the number of directions with significant variance. Best viewed in color.

3.3. Consistency Regularization with Mixup

Even though FDA and PDD produce more accurate pseudo labels, the problem of incorrect pseudo labels remains. Just few of these labels can have a substantial adverse effect on model performance as they can reinforce themselves during the subsequent iterations of Self-Training. To address this issue, we propose a consistency regularization of Self-Training with mixup. We aim to achieve two objectives through mixup regularization. First is to reduce memorization of incorrect pseudo labels for which we increase the strength of the mixup interpolation. Second objective is to enforce consistent predictions on boundary samples (which are most likely to be misclassified) in the target domain and improve generalization capability of the model which is achieved implicitly through mixup of random unlabeled target samples and their corresponding pseudo labels. Perturbation by interpolation have been proven effective for consistency regularization in Semi-Supervised Learning settings [31]. In case of Self-Training, mixup operation functions as a regularizer which enforces consistent predictions on unlabeled samples even after it has been augmented through interpolation.

Mixup operation, as described in [35, 31], for a pair of target samples \((x_i^t, y_i^t)\) and \((x_i'^t, y_i'^t)\) is as follows:

\[
\hat{x}_t = \lambda x_t^t + (1 - \lambda) x_t'^t \\
\hat{y}_t = \lambda y_t^t + (1 - \lambda) y_t'^t,
\]  

where \(\lambda\) is a mixup factor which, following [35], is drawn from the \(Beta(\alpha, \alpha)\) distribution. As discussed before, to minimize the effect of incorrect pseudo labels we choose higher value of \(\alpha\) to increase the strength of the interpolation. Value of \(\alpha\) is a hyper-parameter, which we set to 16.0 in all our experiments as it gives the best results. Here, it should be noted that the pseudo label \(y_t = \{(e_i, l_i)\}_{i=1}^n\) of \(x_t\) have \(m\) class labels corresponding to \(m\) objects. Therefore, unlike mixup operations in classification task [35, 30] where mixup is applied on image labels, we apply mixup operation on object labels. Further, mixup is applied only on the class labels of the objects and not on location labels.

3.4. Self-Training with Mixup (STM)

For the unlabeled target dataset, Self-Training enables learning representations directly from the dataset with the help of pseudo labels. During Self-Training we apply random mixup in which mixup factor \(\lambda\) is drawn randomly from \(Beta(\alpha, \alpha)\) distribution for each iteration. Further, instead of applying mixup only on a pair of target inputs \((x_t^t, x_t'^t)\), following [30], we apply mixup on both inputs and their hidden representations (2 (b)). These hidden representations are also selected randomly following [30]. During Self-Training pseudo labels are periodically updated at every epoch of the target dataset as the model gets better due to the effect of FDA and Self-Training. Finally, with pseudo label mixup the objective of the Self-Training is defined as follows:

\[
\mathcal{L}_{STM} = -\frac{1}{N} \sum_{i=1}^{N} \hat{y}_t^i \log(\hat{y}_t^i) - (1 - \hat{y}_t^i) \log(1 - \hat{y}_t^i)
\]  

4193
where $\tilde{y}_i = p(y|x_i)$ is a probability distribution of the class prediction produced by the model on the input $x_i$ and $N$ is the number of samples in a mini-batch. Please see our supplemental material for the discussion on using localization loss to regress bounding boxes during Self-Training.

### 3.5. Model Training

The training of the object detector happens in two phases. In the first phase, the model is trained with FDA to reduce the distribution shifts between the source and the target datasets. Once the model has been stabilized after training it for a certain number of iterations, in the second phase we enable the Self-Training. In all our experiments, Self-Training runs for 10k to 20k iterations. The overall objective of the training is as follows:

$$L = L_{det} + \lambda_{FDA} L_{FDA} + \lambda_{STM} L_{STM}$$

(6)

where $\lambda_{STM}$ is set to 1 only during the Self-Training otherwise set to 0, $L_{det}$ is SSD [17] objective on the source dataset, $L_{FDA}$ is FDA loss as described in Equation 1, $\lambda_{FDA}$ is a domain adaptation factor which gradually changes from 0 to 1 during training, as described in [7].

### 4. Experiments

In this section we describe experiments and results of the proposed method on three domain adaptation benchmarks for object detection. We also present the effectiveness of the proposed method in real-world domain adaptation scenario for the application of pedestrian detection.

#### 4.1. Implementation

For all experiments, we use one-stage object detector SSD as a baseline detector. The baseline detector employs a VGG16 [28] as a feature extractor pre-trained on the ImageNet [24]. The input images are resized to 300x300. It is trained with Stochastic Gradient Descent (SGD) optimizer, with a momentum of 0.9 and a weight decay of 0.0005. The base learning rate is set to 0.01 with the batch size of 8. For all experiments we train the model for 120k iterations.

For training the model with FDA, global level features are extracted from the first detection head of SSD which is at conv4_3 layer in case of VGG16. These features are fed to the global domain discriminator. The local level features are extracted from shallow layers conv2_2 and conv3_3 of VGG16 and fed to local domain discriminators. To apply the adversarial training strategy, Gradient Reversal Layer (GRL) is added before each of the domain discriminators.

Self-Training is enabled between 70k-90k iterations of training. During Self-Training pseudo labels are updated at every epoch of the target dataset. In data distillation, we set $k = 5$ to ensemble predictions from original image and 4 geometric transformations on the same image. These transformations are horizontal-flip, vertical-flip, down-scale, and up-scale.

#### 4.2. Datasets

For adaptation, PASCAL VOC [6] has been used as the source domain dataset. This dataset has object class annotations of 20 classes and their bounding boxes. We use PASCAL VOC 2007 and 2012 training and validation splits for training. The target domain consists of Clipart1k [10], Comic2k [10] and Watercolor2k [10] datasets. Clipart1k contains 1k images in total, which have the same 20 categories as PASCAL VOC. 500 images were used for training and 500 for testing. Watercolor2k and Comic2k contains 6 categories in common with PASCAL and 2k images in total. 1k training images were utilized during training and 1k images are used as test images.

To test the effectiveness of our proposed method in real-world scenario, we also show results of domain adaptation from COCO [16] to Caltech [5] for the task of pedestrian detection. Caltech dataset is a pedestrian detection benchmark dataset having 16k train samples and 2.24k test samples. For the task of pedestrian detection, detectors are trained only on a person class therefore we use images having person class from the COCO, resulting in around 45k train samples. For more details on Caltech dataset, please see our supplemental material.

#### 4.3. Results

We employ COCO evaluation [16] protocol to measure the mAP of the model, where we set the confidence threshold to 0.05 and the IoU threshold to 0.5. We compare the proposed method with the Source only model [17] which is trained only on the source dataset, DANN [7], Background Score Regularization and Weak Self-Training [14] (denoted
Table 1. Results on the Clipart1k test dataset and comparison of various methods in terms of mAP (%). To adapt SSD [17] on Clipart1k, PASCAL VOC is used as a source dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>mbike</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Only [17]</td>
<td>27.3</td>
<td>60.4</td>
<td>17.5</td>
<td>16.0</td>
<td>14.5</td>
<td>43.7</td>
<td>32.0</td>
<td>10.2</td>
<td>8.6</td>
<td>15.3</td>
<td>24.5</td>
<td>16.0</td>
<td>18.4</td>
<td>49.5</td>
<td>30.7</td>
<td>30.0</td>
<td>2.3</td>
<td>23.0</td>
<td>35.1</td>
<td>29.9</td>
</tr>
<tr>
<td>DANN [7]</td>
<td>24.1</td>
<td>52.6</td>
<td>27.5</td>
<td>18.5</td>
<td>20.3</td>
<td>59.3</td>
<td>37.4</td>
<td>3.8</td>
<td>35.1</td>
<td>32.6</td>
<td>23.9</td>
<td>13.8</td>
<td>22.5</td>
<td>50.9</td>
<td>49.9</td>
<td>36.3</td>
<td>11.6</td>
<td>31.3</td>
<td>48.0</td>
<td>35.8</td>
</tr>
<tr>
<td>BSR+WST [14]</td>
<td>28.0</td>
<td>64.5</td>
<td>23.9</td>
<td>19.0</td>
<td>21.9</td>
<td>64.3</td>
<td>43.5</td>
<td>16.4</td>
<td>42.2</td>
<td>25.9</td>
<td>30.5</td>
<td>7.9</td>
<td>25.7</td>
<td>67.6</td>
<td>54.5</td>
<td>36.4</td>
<td>10.3</td>
<td>31.2</td>
<td>57.4</td>
<td>43.5</td>
</tr>
<tr>
<td>I2Net [3]</td>
<td><strong>30.0</strong></td>
<td>67.0</td>
<td><strong>32.5</strong></td>
<td><strong>21.8</strong></td>
<td>29.2</td>
<td>62.5</td>
<td>41.3</td>
<td>11.6</td>
<td>37.1</td>
<td>39.4</td>
<td>27.4</td>
<td>19.3</td>
<td>25.0</td>
<td>67.4</td>
<td>55.2</td>
<td><strong>42.9</strong></td>
<td>19.5</td>
<td>36.2</td>
<td>50.7</td>
<td>39.3</td>
</tr>
<tr>
<td>DBG[2]</td>
<td>23.2</td>
<td>65.5</td>
<td>30.1</td>
<td>18.3</td>
<td>24.6</td>
<td><strong>67.6</strong></td>
<td>43.9</td>
<td>15.1</td>
<td>38.7</td>
<td>36.4</td>
<td>31.3</td>
<td>20.2</td>
<td>25.0</td>
<td><strong>74.3</strong></td>
<td>55.1</td>
<td>38.2</td>
<td>12.5</td>
<td><strong>41.0</strong></td>
<td>49.1</td>
<td><strong>43.9</strong></td>
</tr>
<tr>
<td>STM</td>
<td>28.2</td>
<td>65.2</td>
<td>14.8</td>
<td>17.9</td>
<td>3.1</td>
<td>32.4</td>
<td>39.1</td>
<td>5.8</td>
<td>45.4</td>
<td>7.8</td>
<td>35.5</td>
<td>15.8</td>
<td>23.8</td>
<td>44.5</td>
<td>55.2</td>
<td>36.4</td>
<td>5.0</td>
<td>23.6</td>
<td>35.1</td>
<td>37.7</td>
</tr>
<tr>
<td>FDA</td>
<td>28.5</td>
<td>63.1</td>
<td>24.4</td>
<td>22.0</td>
<td>18.8</td>
<td>35.1</td>
<td>38.0</td>
<td>2.5</td>
<td>47.7</td>
<td>43.5</td>
<td>23.7</td>
<td>8.4</td>
<td>31.3</td>
<td>49.4</td>
<td>50.3</td>
<td>32.9</td>
<td>12.7</td>
<td>32.9</td>
<td>35.0</td>
<td>39.4</td>
</tr>
<tr>
<td>FDA+STM</td>
<td><strong>26.9</strong></td>
<td><strong>76.7</strong></td>
<td><strong>30.7</strong></td>
<td><strong>34.8</strong></td>
<td><strong>32.9</strong></td>
<td><strong>52.0</strong></td>
<td><strong>49.5</strong></td>
<td>6.3</td>
<td>46.8</td>
<td><strong>55.4</strong></td>
<td><strong>45.1</strong></td>
<td><strong>21.5</strong></td>
<td><strong>41.0</strong></td>
<td><strong>66.3</strong></td>
<td><strong>70.0</strong></td>
<td>42.0</td>
<td>11.7</td>
<td><strong>26.4</strong></td>
<td><strong>51.8</strong></td>
<td><strong>43.8</strong></td>
</tr>
</tbody>
</table>

Table 2. Results on Comic2k test dataset in terms of mAP(%).

<table>
<thead>
<tr>
<th>Method</th>
<th>bike</th>
<th>bird</th>
<th>cat</th>
<th>cat</th>
<th>dog</th>
<th>person</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Only [17]</td>
<td>43.3</td>
<td>9.4</td>
<td>23.6</td>
<td>9.8</td>
<td>10.9</td>
<td>34.2</td>
<td>21.9</td>
</tr>
<tr>
<td>DANN [7]</td>
<td>33.3</td>
<td>11.3</td>
<td>19.7</td>
<td>13.4</td>
<td>19.6</td>
<td>37.4</td>
<td>22.5</td>
</tr>
<tr>
<td>BSR+WST [14]</td>
<td>50.6</td>
<td>13.6</td>
<td>31.0</td>
<td>7.5</td>
<td>16.4</td>
<td>41.4</td>
<td>26.8</td>
</tr>
<tr>
<td>I2Net [3]</td>
<td>47.5</td>
<td><strong>19.9</strong></td>
<td><strong>33.2</strong></td>
<td>11.4</td>
<td>19.4</td>
<td>49.1</td>
<td>30.1</td>
</tr>
<tr>
<td>DBG[2]</td>
<td>45.1</td>
<td>15.9</td>
<td>24.8</td>
<td>11.5</td>
<td>29.4</td>
<td>55.1</td>
<td>30.4</td>
</tr>
<tr>
<td>STM</td>
<td>42.2</td>
<td>11.2</td>
<td>15.5</td>
<td>17.5</td>
<td>8.7</td>
<td>40.1</td>
<td>22.5</td>
</tr>
<tr>
<td>FDA</td>
<td>51.1</td>
<td>15.9</td>
<td>12.1</td>
<td>24.8</td>
<td>25.3</td>
<td>50.4</td>
<td>29.9</td>
</tr>
<tr>
<td>FDA+STM</td>
<td>51.8</td>
<td>16.4</td>
<td>13.1</td>
<td>34.6</td>
<td>32.6</td>
<td>63.8</td>
<td>35.4</td>
</tr>
</tbody>
</table>

Table 3. Results on Watercolor2k test dataset in terms of mAP(%).

<table>
<thead>
<tr>
<th>Method</th>
<th>bike</th>
<th>bird</th>
<th>cat</th>
<th>cat</th>
<th>dog</th>
<th>person</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Only [17]</td>
<td>77.5</td>
<td>46.1</td>
<td>44.6</td>
<td>30.0</td>
<td>26.0</td>
<td>58.6</td>
<td>47.1</td>
</tr>
<tr>
<td>DANN [7]</td>
<td>73.4</td>
<td>41.0</td>
<td>32.4</td>
<td>28.6</td>
<td>22.1</td>
<td>51.4</td>
<td>41.5</td>
</tr>
<tr>
<td>BSR+WST [14]</td>
<td>75.6</td>
<td>45.8</td>
<td>49.3</td>
<td>34.1</td>
<td>30.3</td>
<td>64.1</td>
<td>49.9</td>
</tr>
<tr>
<td>I2Net [3]</td>
<td>81.1</td>
<td>49.3</td>
<td>46.2</td>
<td>35.0</td>
<td>31.9</td>
<td>65.7</td>
<td>51.5</td>
</tr>
<tr>
<td>DBG[2]</td>
<td>84.0</td>
<td>46.7</td>
<td>45.5</td>
<td>36.2</td>
<td>35.7</td>
<td>63.7</td>
<td>52.0</td>
</tr>
<tr>
<td>STM</td>
<td>91.8</td>
<td>50.6</td>
<td>34.4</td>
<td>47.1</td>
<td>37.3</td>
<td>72.7</td>
<td>55.7</td>
</tr>
<tr>
<td>FDA</td>
<td>83.1</td>
<td>46.6</td>
<td>29.5</td>
<td>37.0</td>
<td>32.6</td>
<td>67.4</td>
<td>49.4</td>
</tr>
<tr>
<td>FDA+STM</td>
<td>99.3</td>
<td>52.7</td>
<td>44.5</td>
<td><strong>49.1</strong></td>
<td>38.8</td>
<td><strong>73.3</strong></td>
<td>59.6</td>
</tr>
</tbody>
</table>

Table 4. Results on Caltech dataset for the task of pedestrian detection in terms of mAP (%).

<table>
<thead>
<tr>
<th>Method</th>
<th>Person</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Only [17]</td>
<td>27.1</td>
</tr>
<tr>
<td>DANN [7]</td>
<td>26.8</td>
</tr>
<tr>
<td>STM</td>
<td><strong>32.7</strong></td>
</tr>
<tr>
<td>FDA</td>
<td>28.6</td>
</tr>
<tr>
<td>FDA+STM</td>
<td>32.4</td>
</tr>
</tbody>
</table>

as BSR+WST, I2Net [3] and DBG[2]. BSR+WST, I2Net and DBG are implemented on SSD and have evaluation protocol identical to ours. The results of BSR and BSR+WST are cited from [14], results of I2Net are cited from the original paper. We also demonstrate the effectiveness of individual components of the proposed method which are Feature Distribution Alignment (FDA) and Self-Training with mixup regularization (STM). They are denoted as FDA and STM in the result tables. In results, FDA+STM indicates both components are used in training the model.

**Clipart1k.** As shown in Table 1 the mAP of the proposed FDA+STM is 41.6%, which is an improvement of +14.9% over the Source only model. When only FDA is applied the mAP is 32.0% and applying STM improves the performance of the model by +9.6%. This is a significant improvement and shows the effectiveness of the proposed Self-Training method. The proposed method also outperforms BSR+WST [14], I2Net [3] and DBG[2] by considerable margin. Further, it can be observed that even when STM is applied without FDA, mAP is improved. This shows that mixup is effective in regularizing the Self-Training and it reduces the adverse effect of incorrect pseudo labels.

**Comic2k.** As shown in Table 2, on Comic2k dataset mAP of FDA+STM is 35.4%. In comparison to BSR+WST [14], I2Net [3] and DBG[2], the proposed method outperforms them by a significant margin of +8.6%, +5.3% and +5% respectively. Overall improvement in mAP is 13.5% over the Source only model.

**Watercolor2k.** Results on Watercolor2k dataset are shown in Table 3. When only FDA is applied, there is a marginal improvement in mAP. After employing STM along with FDA, the mAP is further increased by +10.2%. We also observe that when only STM is applied, the mAP is improved by +8.6% over the Source only model. It suggests that when FDA is not very effective due to high intra-domain variance, STM is helpful to boost the performance. The proposed method significantly outperforms state-of-the-art BSR+WST [14], I2Net [3] and DBG[2] method and achieve the best mAP.

**Caltech Pedestrian Benchmark.** In this case, we examine the effectiveness of the proposed method in real-world domain adaptation scenario. We use COCO [16] dataset as a source domain and train SSD [17] on it to detect objects of only a person class. The mAP obtained is 61.6%. When the test dataset of Caltech is evaluated with the Source only model, it gives a mAP of 27.1%. COCO is a large-scale dataset in which images are collected from various sources and have high intra-domain variance. Therefore, it can be observed in Table 4, that the adaptation by FDA is not helpful. Employing STM along with FDA is very effective and improves a mAP by +5.3% to 32.4%. These results show that the STM is effective in challenging real-world scenarios too.
Table 5. mAP (%) of target test sets for different mixup strength.

<table>
<thead>
<tr>
<th>Method</th>
<th>Clipart1k</th>
<th>Comic2k</th>
<th>Watercolor2k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Only[17]</td>
<td>26.7</td>
<td>21.9</td>
<td>47.1</td>
</tr>
<tr>
<td>FDA+STM ($\alpha = 2.0$)</td>
<td>40.0</td>
<td>32.4</td>
<td>54.0</td>
</tr>
<tr>
<td>FDA+STM ($\alpha = 4.0$)</td>
<td>39.9</td>
<td>31.6</td>
<td>57.1</td>
</tr>
<tr>
<td>FDA+STM ($\alpha = 8.0$)</td>
<td>40.2</td>
<td>35.4</td>
<td>57.4</td>
</tr>
<tr>
<td>FDA+STM ($\alpha = 16.0$)</td>
<td>41.6</td>
<td>35.4</td>
<td>59.6</td>
</tr>
</tbody>
</table>

4.4. Discussion and Ablation Studies

In this section, we present the analysis of the proposed method. We conducted the ablation studies on Self-Training, analyze the hyper-parameter sensitivity of $\alpha$ which controls the mixup ratio, and study the effect of disabling data distillation during Self-Training.

**Effect of mixup regularization during Self-Training.** We investigate the effectiveness of mixup regularization during Self-Training. While training SSD with FDA and Self-Training on adaptation from PASCAL VOC to Clipart1k, we measure the mAP of the model on Clipart1k at iterations during Self-Training period. As we can see in Figure 4, with mixup regularization, Self-Training is highly stable without adverse impact on the performance even when it runs for very long duration. In contrast, without mixup regularization during Self-Training, the performance improves momentarily for few iterations and then it rapidly degrades due to reinforcement of incorrect pseudo labels. This shows mixup is useful in minimizing the impact of incorrect pseudo labels.

**Effect of mixup parameter $\alpha$.** The hyper-parameter $\alpha$ controls the mixup ratio drawn from $\lambda \sim Beta(\alpha, \alpha)$ distribution. Increasing the value of $\alpha$ will increase the strength of interpolation and will generate the interpolated samples further from the real samples. In effect, if labels of such samples are random or corrupt then it becomes difficult for the network to learn such associations. Thus, minimizing the effect of incorrect pseudo labels. We trained the model with different values of mixup $\alpha$ and results are shown in Table 5 on Clipart1k, Comic2k and Watercolor2k datasets.

**Effect of Periodic Data Distillation.** To analyze the effectiveness of PDD, we disable it and train the model with FDA and STM on Clipart1k, Comic2k and Watercolor2k datasets. In this experiment, we do not use multi-transform inference on the model to generate pseudo labels and predictions only from the original target samples are used as pseudo labels. As shown in Figure 5, we observe that disabling the data distillation reduces the performance of the model on all three benchmarks. It shows that Periodic Data Distillation is beneficial in generating more accurate pseudo labels and helps in achieving the optimal performance.

**Feature Visualization.** Figure 3 (a) presents the visualization of t-SNE [29] feature representations of two classes learned by Baseline and FDA+STM model. Baseline model has lower discriminative ability on target domain classes.

![Figure 5. Comparison of FDA+STM mAP(%) in two different settings. FDA+STM with PDD (orange bars) outperforms FDA+STM without PDD (blue bars) on all three benchmarks.](image)

Feature representations at both conv4_3 and conv7 layers have overlapping space due to which detector has low classification accuracy. On the other hand, after Self-Training with mixup regularization, both classes have a separate feature space which improves discriminative capability of the model on target domain.

**Reducing directions of variance.** In Figure 3 (b) we analyze flattening property of mixup regularization. Authors of [35, 30] show that mixup regularization leads to flattening: a reduction in the number of directions with variability [30]. Reducing directions of variability improves object classification ability of the detector. To investigate flattening, we compute Singular Value Decomposition of class specific representation at layer conv10_2. As compared to Baseline, in FDA+STM, we observe that large singular values are concentrated at lower indices which means that class specific representation has lower variance.

5. Conclusion

In this paper, we proposed a domain adaptation method using FDA and Self-Training for one-stage object detector. Our method consists of two major components. First is FDA, which utilizes adversarial loss to reduce the domain shifts between source and target domain. As domain shifts reduce, the model gets better at predictions on target domain samples. Taking advantage of this property, we use PDD to generate more accurate pseudo labels of the target domain. In the second component, STM, we use pseudo labels to guide the Self-Training of an object detector and regularize Self-Training with mixup. These two enable a stable and effective Self-Training and achieve state-of-the-art performance on domain adaptation benchmarks. Further, we validate the application of the proposed method in real-world scenario for the task of pedestrian detection where it shows a significant improvement in the performance.
References


