IFMix: Utilizing Intermediate Filtered Images for Domain Adaptation in Classification

Saeed Bakhshi Germi[a] and Esa Rahtu[b]
Computer Vision Group, Tampere University, Tampere, Finland

Keywords: Domain Adaptation, Filtered Images, Classification, Mixup Technique.

Abstract: This paper proposes an iterative intermediate domain generation method using low- and high-pass filters. Domain shift is one of the prime reasons for the poor generalization of trained models in most real-life applications. In a typical case, the target domain differs from the source domain due to either controllable factors (e.g., different sensors) or uncontrollable factors (e.g., weather conditions). Domain adaptation methods bridge this gap by training a domain-invariant network. However, a significant gap between the source and the target domains would still result in bad performance. Gradual domain adaptation methods utilize intermediate domains that gradually shift from the source to the target domain to counter the effect of the significant gap. Still, the assumption of having sufficiently large intermediate domains at hand for any given task is hard to fulfill in real-life scenarios. The proposed method utilizes low- and high-pass filters to create two distinct representations of a single sample. After that, the filtered samples from two domains are mixed with a dynamic ratio to create intermediate domains, which are used to train two separate models in parallel. The final output is obtained by averaging out both models. The method's effectiveness is demonstrated with extensive experiments on public benchmark datasets: Office-31, Office-Home, and VisDa-2017. The empirical evaluation suggests that the proposed method performs better than the current state-of-the-art works.

1 INTRODUCTION

With the increasing popularity of deep learning algorithms in the heavy machine industry and the inclusion of artificial intelligence in new regulations (e.g., EU AI Act) and safety standards (e.g., ISO/IEC JTC 1/SC 42 Committee), the practical issues of utilizing such algorithms in safety-critical applications have become more apparent. One of the challenges for any practical application of a deep learning algorithm is collecting and labeling a large dataset for training the algorithm while considering the safety criteria for the application (Bakhshi Germi and Rahtu, 2022b). A standard method to deal with this issue is utilizing transfer learning (Zhuang et al., 2021), where the model is trained with a label-rich source dataset (e.g., synthesized or simulated data) and fine-tuned on a much smaller target dataset (e.g., data collected from the real world). However, a significant gap between these two domains would result in poor performance.

Gradual domain adaptation (GDA) deals with the gap problem by adding data from intermediate domains that interpolate between the source and the target domains (Kumar et al., 2020). The intermediate domains are assumed to be available with sufficient data for the training process. The accuracy of GDA methods is highly dependent on the distance between the source and the target domains. Moreover, GDA methods are usually unsupervised and do not require labels from intermediate or target domains. While unsupervised methods attract more attention in the research community, using a small labeled subset from the target domain is more realistic in real-world applications. Various annotation tools (Adhikari and Huttunen, 2021) and denoising techniques (Bakhshi Germi and Rahtu, 2022a) could be utilized to help with gathering the required labeled subset. Meanwhile, intermediate domains do not naturally exist for most real-world applications. Thus, this paper focuses on generating intermediate domains based on a large labeled source dataset and a small labeled target dataset.

This paper proposes IFMix, a domain adaptation algorithm that utilizes a filtered-image-based mixup technique to create intermediate domains iteratively. A new domain is created by merging the low-pass or high-pass filtered images from both domains with a dynamic ratio. The images are chosen from the same

[a] https://orcid.org/0000-0003-3048-220X
[b] https://orcid.org/0000-0001-8767-0864

Germi, S. and Rahtu, E.
IFMix: Utilizing Intermediate Filtered Images for Domain Adaptation in Classification.
DOI: 10.5220/0011713600003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 5: VISAPP*, pages 205-211
ISBN: 978-989-758-634-7; ISSN: 2184-4321

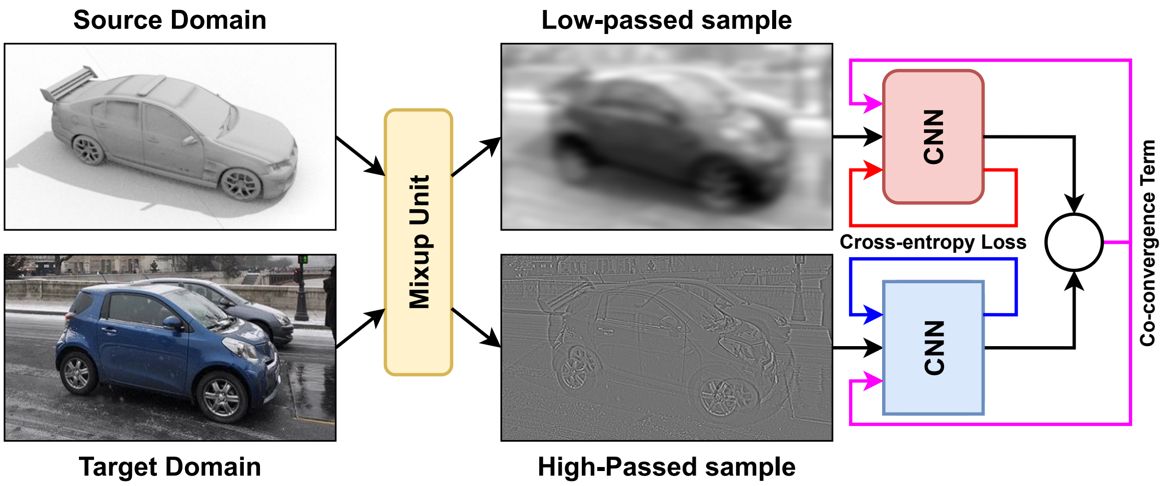

Figure 1: The overall structure of the proposed method. Two samples of the same category are chosen from two domains to be mixed. The mixup unit utilizes low-pass and high-pass filters to mix images with different ratios. The resulting images are used as training samples for two separate models. Each model is trained with a categorical cross-entropy loss. A co-convergence term is utilized to ensure the convergence of both models towards the same point.

category in both domains to keep the labels intact. After that, the proposed method utilizes the intermediate domains to train two separate models in parallel. Both models' average output is considered the proposed method's final output. The intuition behind the proposed method is that a supervised method that relies on a small amount of data from the target domain would be practical and realistic, the iterative domain creation would compensate for the lack of data in real-world applications, and the two models develop different perspectives based on their respective filters.

The main difference between the proposed method and previous works is utilizing a small labeled target dataset to create intermediate domains, resulting in accurate labels instead of pseudo-labels. Also, using the low- and high-pass filters would result in two distinct representations of the same sample, creating substantially different intermediate domains for training two different models. Moreover, the iterative and gradual nature of the algorithm ensures that the model is not overwhelmed by new information while the gap between the two domains is breached. The effectiveness of the proposed method is shown by comparing the performance with previous state-of-the-art methods in standard public benchmarks such as Office-31 (Saenko et al., 2010), Office-Home (Venkateswara et al., 2017), and VisDa-2017 (Peng et al., 2017). The main contributions of this paper are summarized as follows:

- Proposing an iterative intermediate domain creation technique based on filtered images to bridge the gap between the source and the target domains.
- Providing a practical domain adaptation algorithm based on the proposed intermediate domains.
- Providing empirical evaluation with extensive experiments on three standard benchmarks to show the effectiveness of the proposed method.

The rest of the paper is structured as follows. Section 2 covers the related works. Next, Section 3 explains the proposed method in detail. After that, Section 4 deals with the experiments and the empirical evaluation to show the effectiveness of the proposed method. Finally, Section 5 concludes the work.

2 RELATED WORKS

2.1 Unsupervised Domain Adaptation

Unsupervised domain adaptation (UDA) methods utilize domain-invariant representation to generalize a model from a rich-labeled source domain to an unlabeled target domain (Wilson and Cook, 2020). The process can be done by either optimizing distribution discrepancy metrics (e.g., maximum mean discrepancy) (Li et al., 2021a; Peng et al., 2019) or utilizing adversarial training (Li et al., 2021b; Liu et al., 2019; Wang et al., 2019). On top of that, utilizing pseudo-labeling ideas from semi-supervised learning methods improves the performance of UDA algorithms (Chen et al., 2020; Liang et al., 2020; Liang et al., 2021; Liu et al., 2021; Zhang et al., 2021b). Moreover, the natural advantage of transformers in extracting transferable representations was studied further for application in domain adaptation (Ma et al.,

Figure 2: The creation of multiple intermediate domains by the proposed method. The shown samples are not filtered to understand better how the method works. Samples progress from the source domain (left) to the target domain (right) with each iteration based on the value of H.

2021; Xu et al., 2021; Yang et al., 2021a). Unsupervised methods have been the research focus for a while in academic applications. However, utilizing a small labeled dataset could result in a performance surge without significantly increasing the overall cost of gathering data.

2.2 Gradual Domain Adaptation

Gradual domain adaptation methods utilize intermediate domains to improve the performance of basic domain adaptation techniques (Choi et al., 2020; Cui et al., 2020; Dai et al., 2021; Hsu et al., 2020). GDA methods utilize generative models (e.g., generative adversarial networks) to create an intermediate domain by mixing the source and the target data at an arbitrary ratio (Sagawa and Hino, 2022). By doing so, the model can learn common features shared between two domains. In the original work, Kumar assumed that the intermediate domains gradually shift from the source to the target domain, and their sequence is known prior to learning (Kumar et al., 2020). However, the method is effective even if the sequence of these domains is unknown (Chen and Chao, 2021; Zhang et al., 2021a) or when no intermediate domain is available (Abnar et al., 2021; Na et al., 2021b). The main difference between current state-of-the-art GDA algorithms is their technique for creating intermediate domains.

2.3 Mixup Technique

Mixup techniques are a family of data augmentation methods based on mixing two or more data points. Mixup and its variants have proven helpful in supervised and semi-supervised learning (Berthelot et al., 2019; Yun et al., 2019; Zhang et al., 2017). Some recent domain adaptation methods tried utilizing this technique to create a continuous latent space across domains (Wu et al., 2020; Xu et al., 2020), obtain pseudo labels for intermediate domains (Na et al., 2021b; Yan et al., 2020; Yang et al., 2021b), or generate more positive/negative samples (Kalantidis et al., 2020; Zhang et al., 2022; Zhu et al., 2021).

This paper utilizes the intermediate domains from GDA, a mixup technique based on low- and high-pass filters, and a small labeled subset from the target domain to achieve high performance in real-world scenarios. The assumptions in this paper are tailored around practical use cases of domain adaptation where a large labeled source domain and a small labeled target domain are available. While similar works exist in this field, the proposed method outperforms the existing state-of-the-art, as shown in Section 4.

3 PROPOSED METHOD

This section presents the details of the proposed method, as shown in Figure 1. Let $\mathcal{D}^s = \{(x_i^s, y_i^s)\}_{i=1}^n$ be the labeled dataset from the source domain, $\mathcal{D}^t = \{(x_j^t)\}_{j=1}^m$ be the unlabeled dataset from the target domain, and $\mathcal{D}_l^t = \{(x_k^t, y_k^t)\}_{k=1}^p$ be the labeled subset from the target domain. The task is transferring knowledge from $\mathcal{D}^s$ to $\mathcal{D}^t$ when there is a large distribution gap between them.

3.1 Iterative Filtered Mixup

The proposed method selects random samples with the same category label from $\mathcal{D}^s$ and $\mathcal{D}_l^t$, applies low- and high-pass filters on them, and mixes them to create new samples as follows:

$$
\begin{aligned}
x_i^{lo} &= (1-H) \times LoPass(x_i^s) + H \times LoPass(x_j^t) \\
x_i^{hi} &= (1-H) \times HiPass(x_i^s) + H \times HiPass(x_j^t)
\end{aligned} \tag{1}
$$

Where ($0 \leq H \leq 1$) denotes a dynamic ratio for the mixing step, $LoPass$ and $HiPass$ denote the low-pass and high-pass filter functions, respectively. These filters could be implemented using the Gaussian filter function in the Multidimensional Image Processing package (scipy.ndimage). Moreover, the labels y_i^{lo} and y_i^{hi} for generated samples would be the same as the original label y_i due to choosing samples from the same category. Finally, the mixing ratio H is updated based on the number of epochs as follows:

$$
H_{i+1} = H_i + \alpha \times t \tag{2}
$$

Where α is a positive constant and t is the current number of epochs. Two labeled datasets, $\mathcal{D}_H^{lo}$ and $\mathcal{D}_H^{hi}$, are created with each iteration. These intermediate datasets fill the gap between the source and the target domains, as shown in Figure 2. Note that the figure shows unfiltered samples for a more straightforward interpretation of how the algorithm works.

3.2 Training and Loss Functions

In the next step, two models are trained on $\mathcal{D}_H^{lo}$ and $\mathcal{D}_H^{hi}$ using the categorical cross-entropy loss function:

$$
\begin{aligned}
\mathcal{L}_{cce}^{lo} &= \frac{1}{B}\sum_{i}^{B} y_i^{lo} \times log\left(p\left(y|x_i^{lo}\right)\right) \\
\mathcal{L}_{cce}^{hi} &= \frac{1}{B}\sum_{i}^{B} y_i^{hi} \times log\left(q\left(y|x_i^{hi}\right)\right)
\end{aligned} \tag{3}
$$

Where $p(y|x_i^{lo})$ and $q(y|x_i^{hi})$ denote the predicted class for each network on their respective input, and B is the batch size. The models are trained separately for a few epochs (warm-up period) to ensure they gain different perspectives without the influence of the other model.

3.3 Output and Co-Convergence Term

With each model training to recognize different characteristics of a given sample, their average output is used to determine the final output of the algorithm. Since the models should converge towards the same goal, a co-convergence term is added to the overall loss after the warm-up period. This term ensures that each model can influence the other model slightly to reach a similar conclusion on their output.

$$
\mathcal{L}_{cct} = \frac{1}{B}\sum_{i}^{B} y_i \times log\left(\frac{p\left(y|x_i^{lo}\right) + q\left(y|x_i^{hi}\right)}{2}\right) \tag{4}
$$

3.4 Overall Process

The overall process of the IFMix algorithm is summarized in Algorithm 1. The algorithm starts with creating the intermediate domains in each iteration. Then two networks are trained with the new intermediate domains using the defined loss functions. The co-convergence term is added after the warm-up period to allow the models to develop unique characteristics without the influence of the other model.

In experiments, the mixup ratio H is updated every few epochs to prevent potential divergence of models.

Algorithm 1: IFMix Algorithm.

```
Require: Source dataset D^s, Labeled Target
    subset D_l^t, Number of epochs T, Batch size
    B, Warm-up period W, Mixup ratio H,
    Mixup increment rate α
for t ∈ 1,...,T do
    Select samples from same category in D^s
    and D_l^t
    Create intermediate domains D_H^hi and D_H^lo
    using Eq. 1
    for b ∈ 1,...,B do
        Update loss functions L_cce^lo and L_cce^hi
        using Eq. 3
        if i ≥ W then
            Update co-convergence term L_cct
            using Eq. 4
        end if
    end for
    Update the mixup ratio
    using Eq. 2
end for
```

4 EXPERIMENTS & EVALUATION

To evaluate the proposed method, three different domain adaptation benchmarks are chosen so that the performance of the proposed method can be compared with state-of-the-art methods. In each experiment, 5% of samples from the target domain are selected as labeled target subsets for the proposed method, and the remaining 95% of samples are left as test data.

4.1 Office-31

Office-31 (Saenko et al., 2010), a domain adaptation benchmark, provides samples for 31 categories from three domains. These domains are denoted as A for images taken from Amazon.com, D for images taken with a DSLR camera, and W for images taken with a webcam. The dataset has around 4000 samples, making it a perfect benchmark for proof of concept.

4.2 Office-Home

Office-Home (Venkateswara et al., 2017), a domain adaptation benchmark, provides samples for 65 categories from four domains. These domains are denoted as A for arts and paintings, C for clipart images, P for product images without a background, and R for real-

Table 1: Accuracy (%) on the Office-31 dataset. The best accuracy is indicated in bold, and the second best is underlined.

Method	$A \to D$	$A \to W$	$D \to A$	$D \to W$	$W \to A$	$W \to D$	Average
GSDA (Hu et al., 2020)	94.8	95.7	73.5	99.1	74.9	**100**	89.7
SRDC (Tang et al., 2020)	95.8	95.7	76.7	99.2	77.1	**100**	90.8
RSDA (Gu et al., 2020)	95.8	96.1	77.4	**99.3**	78.9	**100**	91.1
FixBi (Na et al., 2021b)	95	96.1	**78.7**	**99.3**	79.4	**100**	91.4
CoVi (Na et al., 2021a)	**98**	**97.6**	77.5	**99.3**	78.4	**100**	91.8
IFMix (Ours)	97.6	97.5	77.9	**99.3**	**79.7**	**100**	**92**

Table 2: Accuracy (%) on the Office-Home dataset. The best accuracy is indicated in bold, and the second best is underlined.

Method	$A \to C$	$A \to P$	$A \to R$	$C \to A$	$C \to P$	$C \to R$	$P \to A$	$P \to C$	$P \to R$	$R \to A$	$R \to C$	$R \to P$	Average
MetaAlign (Wei et al., 2021)	59.3	76	80.2	65.7	74.7	75.1	65.7	56.5	81.6	74.1	61.1	85.2	71.3
FixBi (Na et al., 2021b)	58.1	77.3	80.4	67.7	79.5	78.1	65.8	57.9	81.7	76.4	62.9	86.7	72.7
CoVi (Na et al., 2021a)	58.5	78.1	80	68.1	80	77	66.4	60.2	82.1	76.6	**63.6**	86.5	73.1
CDTrans (Xu et al., 2021)	60.6	79.5	82.4	75.6	81	82.3	72.5	56.7	84.4	77	59.1	85.5	74.7
WinTR (Ma et al., 2021)	65.3	**84.1**	85	76.8	**84.5**	84.4	73.4	60	85.7	77.2	63.1	86.8	77.2
IFMix (Ours)	**66.1**	84	**86.6**	**77.4**	84.1	**86.1**	**75.2**	**61.1**	**86.5**	**78.4**	62.8	**87.4**	**78**

Table 3: Accuracy (%) on the VisDa-2017 dataset. The best accuracy is indicated in bold, and the second best is underlined.

Method	Plane	Bike	Bus	Car	Horse	Knife	Motor	Human	Plant	Skate	Train	Truck	Average
CAN (Kang et al., 2019)	97	87.2	82.5	74.3	97.8	96.2	90.8	80.7	96.6	96.3	87.5	59.9	87.2
FixBi (Na et al., 2021b)	96.1	87.8	90.5	90.3	96.8	95.3	92.8	**88.7**	97.2	94.2	90.9	25.7	87.2
CDTrans (Xu et al., 2021)	97.1	90.5	82.4	77.5	96.6	96.1	93.6	88.6	97.9	86.9	90.3	**62.8**	88.4
CoVi (Na et al., 2021a)	96.8	85.6	88.9	88.6	97.8	93.4	91.9	87.6	96	93.8	93.6	48.1	88.5
WinTR (Ma et al., 2021)	**98.7**	91.2	**93**	91.9	98.1	96.1	**94**	72.7	97	**95.5**	**95.3**	57.9	90.1
IFMix (Ours)	98.2	**91.7**	92.9	**92.2**	**98.5**	**96.5**	93.7	88	**98**	**95.5**	94.8	61.8	**91.8**

world images taken with a camera. The dataset has around 15000 samples, making it a more challenging task than Office-31.

4.3 VisDa-2017

VisDa-2017 (Peng et al., 2017), a domain adaptation benchmark, provides samples for 12 categories from two domains, simulated and real-world. The dataset has around 280000 samples, making it a complex and realistic benchmark for domain adaptation problems.

4.4 Hyper-Parameters

In the experiments with Office datasets, ResNet-50 with stochastic gradient descent (SGD) is used as the base model. The initial learning rate is 0.001, the momentum is 0.9, the weight decay is 0.005, the initial mixup ratio is 0.05 with a 0.05 increment every 10 epochs, and the total number of epochs is 200. In the experiments with VisDA dataset, the base model is swapped to ResNet-101. The initial learning rate is 0.0001, the initial mixup ratio is 0.1 with a 0.1 increment every 5 epochs, and the total number of epochs is 50. In all experiments, the models utilize pre-trained weights on ImageNet (Russakovsky et al., 2015).

4.5 Results and Comparison

Table 1 holds the results for the Office-31 dataset. Six different tasks are experimented upon, and the results are compared with state-of-the-art methods. The accuracy of state-of-the-art methods is obtained from their respective published papers. The results from each task indicate that the proposed method is competitive. The average accuracy of the proposed method is 92%, which is a slight improvement over the previous best method, CoVi (Na et al., 2021a). As stated before, the Office-31 dataset was utilized to prove that the proposed method works as intended, even if the improvement is slight and negligible.

Table 2 holds the results for the Office-Home dataset. Twelve different tasks are experimented upon, and the results are compared with state-of-the-art methods. Similar to previous experiments, the accuracy of state-of-the-art methods is obtained from their respective published papers. The results from each task indicate that the proposed method is still competitive. The average accuracy of the proposed method is 78%, which is an improvement over the previous best method, WinTR (Ma et al., 2021). Note that the proposed method outperformed CoVi (Na et al., 2021a), the previous best method on the Office-31 dataset, by 4.9% on average. This experiment offers more insight into the value of utilizing the proposed method. While the proposed method slightly outperforms the alternatives in this case, it also offers a more robust solution that works on different datasets.

Table 3 holds the results for the VisDa-2017 dataset. The results are compared on category and overall level. Similar to previous experiments, the accuracy of state-of-the-art methods is obtained from their respective published papers. The results from each category indicate that the proposed method is operating as intended. The average accuracy of the proposed method is 91.8%, which is a significant improvement over the previous best method, WinTR (Ma et al., 2021). The proposed method offers a noticeable improvement in this experiment.

5 CONCLUSION

This paper proposed a practical domain adaptation method that utilizes a labeled subset from the target domain and low- and high-pass filters to create intermediate domains. The iterative creation of intermediate domains helps the model quickly adapt despite a significant gap between domains. The effectiveness of the proposed method is shown with empirical experiments on public benchmark datasets. The proposed method outperforms the current state-of-the-art methods by a noticeable margin while maintaining robustness over different datasets.

ACKNOWLEDGEMENT

This research is part of a Ph.D. study co-funded by Tampere University and Forum for Intelligent Machines ry (FIMA).

REFERENCES

Abnar, S., Berg, R. v. d., Ghiasi, G., Dehghani, M., Kalchbrenner, N., and Sedghi, H. (2021). Gradual domain adaptation in the wild: When intermediate distributions are absent.

Adhikari, B. and Huttunen, H. (2021). Iterative bounding box annotation for object detection. In *International Conference on Pattern Recognition (ICPR)*.

Bakhshi Germi, S. and Rahtu, E. (2022a). Enhanced data-recalibration: Utilizing validation data to mitigate instance-dependent noise in classification. In *Image Analysis and Processing (ICIAP)*.

Bakhshi Germi, S. and Rahtu, E. (2022b). A practical overview of safety concerns and mitigation methods for visual deep learning algorithms. In *Proceedings of the Workshop on Artificial Intelligence Safety (SafeAI)*.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*.

Chen, H.-Y. and Chao, W.-L. (2021). Gradual domain adaptation without indexed intermediate domains. In *Advances in Neural Information Processing Systems*.

Chen, M., Zhao, S., Liu, H., and Cai, D. (2020). Adversarial-learned loss for domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(4).

Choi, J., Choi, Y., Kim, J., Chang, J., Kwon, I., Gwon, Y., and Min, S. (2020). Visual domain adaptation by consensus-based transfer to intermediate domain. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(7).

Cui, S., Wang, S., Zhuo, J., Su, C., Huang, Q., and Tian, Q. (2020). Gradually vanishing bridge for adversarial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dai, Y., Liu, J., Sun, Y., Tong, Z., Zhang, C., and Duan, L.-Y. (2021). Idm: An intermediate domain module for domain adaptive person re-id. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Gu, X., Sun, J., and Xu, Z. (2020). Spherical space domain adaptation with robust pseudo-label loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hsu, H.-K., Yao, C.-H., Tsai, Y.-H., Hung, W.-C., Tseng, H.-Y., Singh, M., and Yang, M.-H. (2020). Progressive domain adaptation for object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Hu, L., Kan, M., Shan, S., and Chen, X. (2020). Unsupervised domain adaptation with hierarchical gradient synchronization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kalantidis, Y., Sariyildiz, M. B., Pion, N., Weinzaepfel, P., and Larlus, D. (2020). Hard negative mixing for contrastive learning. In *Advances in Neural Information Processing Systems*.

Kang, G., Jiang, L., Yang, Y., and Hauptmann, A. G. (2019). Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kumar, A., Ma, T., and Liang, P. (2020). Understanding self-training for gradual domain adaptation. In *Proceedings of the 37th International Conference on Machine Learning (PMLR)*.

Li, S., Liu, C. H., Lin, Q., Wen, Q., Su, L., Huang, G., and Ding, Z. (2021a). Deep residual correction network for partial domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7).

Li, S., Xie, M., Lv, F., Liu, C. H., Liang, J., Qin, C., and Li, W. (2021b). Semantic concentration for domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Liang, J., Hu, D., and Feng, J. (2020). Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of the 37th International Conference on Machine Learning (PMLR)*.

Liang, J., Hu, D., and Feng, J. (2021). Domain adaptation with auxiliary target domain-oriented classifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, H., Long, M., Wang, J., and Jordan, M. (2019). Transferable adversarial training: A general approach to adapting deep classifiers. In *Proceedings of the 36th International Conference on Machine Learning (PMLR)*.

Liu, H., Wang, J., and Long, M. (2021). Cycle self-training for domain adaptation. In *Advances in Neural Information Processing Systems*.

Ma, W., Zhang, J., Li, S., Liu, C. H., Wang, Y., and Li, W. (2021). Exploiting both domain-specific and invariant knowledge via a win-win transformer for unsupervised domain adaptation.

Na, J., Han, D., Chang, H. J., and Hwang, W. (2021a). Contrastive vicinal space for unsupervised domain adaptation.

Na, J., Jung, H., Chang, H. J., and Hwang, W. (2021b). Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. (2019). Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., and Saenko, K. (2017). Visda: The visual domain adaptation challenge.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., and et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3).

Saenko, K., Kulis, B., Fritz, M., and Darrell, T. (2010). Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*.

Sagawa, S. and Hino, H. (2022). Gradual domain adaptation via normalizing flows.

Tang, H., Chen, K., and Jia, K. (2020). Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, X., Li, L., Ye, W., Long, M., and Wang, J. (2019). Transferable attention for domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1).

Wei, G., Lan, C., Zeng, W., and Chen, Z. (2021). Metaalign: Coordinating domain alignment and classification for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wilson, G. and Cook, D. J. (2020). A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology*, 11(5).

Wu, Y., Inkpen, D., and El-Roby, A. (2020). Dual mixup regularized learning for adversarial domain adaptation. In *European Conference on Computer Vision (ECCV)*.

Xu, M., Zhang, J., Ni, B., Li, T., Wang, C., Tian, Q., and Zhang, W. (2020). Adversarial domain adaptation with domain mixup. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(4).

Xu, T., Chen, W., Wang, P., Wang, F., Li, H., and Jin, R. (2021). Cdtrans: Cross-domain transformer for unsupervised domain adaptation.

Yan, S., Song, H., Li, N., Zou, L., and Ren, L. (2020). Improve unsupervised domain adaptation with mixup training.

Yang, G., Tang, H., Zhong, Z., Ding, M., Shao, L., Sebe, N., and Ricci, E. (2021a). Transformer-based source-free domain adaptation.

Yang, L., Wang, Y., Gao, M., Shrivastava, A., Weinberger, K. Q., Chao, W.-L., and Lim, S.-N. (2021b). Deep co-training with task decomposition for semi-supervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017). Mixup: Beyond empirical risk minimization.

Zhang, Y., Deng, B., Jia, K., and Zhang, L. (2021a). Gradual domain adaptation via self-training of auxiliary models.

Zhang, Y., Li, J., and Wang, Z. (2022). Low-confidence samples matter for domain adaptation.

Zhang, Y., Wang, Z., and Mao, Y. (2021b). Rpn prototype alignment for domain adaptive object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhu, R., Zhao, B., Liu, J., Sun, Z., and Chen, C. W. (2021). Improving contrastive learning by visualizing feature transformation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2021). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1).