TRANSFERING LOW-FREQUENCY FEATURES FOR DOMAIN ADAPTATION

Zhaowen Li[1,2], Xu Zhao[1], Chaoyang Zhao[1,3], Ming Tang[1] and Jinqiao Wang[1,2]

[1] National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences,
Beijing, China
[3] Development Research Institute of Guangzhou Smart City
{zhaowen.li,xu.zhao,chaoyang.zhao,tangm,jqwang}@nlpr.ia.ac.cn

ABSTRACT

Previous unsupervised domain adaptation methods did not handle the cross-domain problem from the perspective of frequency for computer vision. The images or feature maps of different domains can be decomposed into the low-frequency component and high-frequency component. This paper proposes the assumption that low-frequency information is more domain-invariant while the high-frequency information contains domain-related information. Hence, we introduce an approach, named low-frequency module (LFM), to extract domain-invariant feature representations. The LFM is constructed with the digital Gaussian low-pass filter. Our method is easy to implement and introduces no extra hyperparameter. We design two effective ways to utilize the LFM for domain adaptation, and our method is complementary to other existing methods and formulated as a plug-and-play unit that can be combined with these methods. Experimental results demonstrate that our LFM outperforms state-of-the-art methods for various computer vision tasks, including image classification and object detection.

Index Terms— domain adaptation, unsupervised, frequency learning

1. INTRODUCTION

Unsupervised domain adaptation (UDA) methods can transfer a learner for the target domain data while manual annotations are only provided in source domain data. The principal idea of UDA methods is to mitigate the domain shift in data distributions. For the UDA image classification task, some previous work [1, 2] minimize the domain-discrepancy to obtain domain-invariant feature representations in convolutional neural networks (CNNs), where the domain-discrepancy is measured by Maximum Mean Discrepancy (MMD) [1] or Joint MMD (JMMD) [2]. Another popular idea for UDA is to adopt an adversarial learning method to obtain domain-invariant features. RevGrad [3] is a representative of these adversarial learning methods by back-propagating the reverse gradients of the domain classifier. These methods mainly focus on learning a global domain shift, aligning the global source and target distributions without considering the category information in both domains. Recently, some researchers considered that making use of pseudo label can help the network to better align domain-invariant features. For example, based on MMD, CAN [4] proposed a contrastive adaptation network, which optimizes contrastive domain discrepancy explicitly modeling the intra-class domain discrepancy and the inter-class domain discrepancy. Additionally, there are some methods that are specifically designed for object detection. The authors of [5] presented two domain adaptation components, image-level adaptation and instance-level adaptation. They adopt domain adversarial approach using a discriminator for each component. The similar motivation was used to align feature representation across domains on enlarged positive regions [6]. Mean Teacher with object relations [7] was also considered, which addressed the adaptive detection from the viewpoint of graph-structured consistency. However, the above methods do not handle the domain adaptation problem from the perspective of frequency.

It is well known [8, 9, 10] that a single natural image or feature map can be decomposed into a low-frequency component that describes the smoothly changing structure, and a high-frequency component that describes the rapidly changing fine details. This paper proposes an assumption that the low-frequency information of the same class in different domains has domain-invariant characteristics. To better present this statement, in Fig. 1, we visualize the distribution of original image data and low-frequency data obtained from original data processed by the digital Gaussian low-pass filter [9]. Fig. 1 illustrates the data distributions of the **W** domain and **A** domain of the Office-31 dataset. The **W** domain and **A** domain consist of the images of the same classes. On the left side of Fig. 1, the domain-discrepancy of the two distinct domains using the whole-frequency information is large. In contrast, on the right, the domain-discrepancy of the two domains

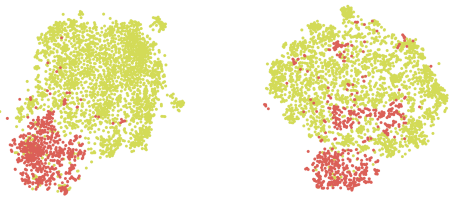

Fig. 1: t-SNE [14] visualization of the distribution of original image data and low-frequency data in the **Amazon (A)** domain and **Webcam (W)** domain of the Office-31 dataset [15]. **Left**: t-SNE of original image data. **Right**: Low-frequency data. The **A** domain consists of 2817 images (yellow point) and **W** domain consists of 795 images (red point).

utilizing the low-frequency information is reduced. Hence, it is reasonable to assume that the low-frequency information is more domain-invariant than the whole-frequency information contained in datasets. Meanwhile, the high-frequency information suppressed by the digital Gaussian low-pass filter contains domain-related information and easily affects the alignment of the data distribution.

To improve the generalization performance of the network, in this paper, we propose a simple yet effective method called low-frequency module (LFM). The LFM is constructed with the digital Gaussian low-pass filter. It can enhance the generalization performance of models by utilizing the inherent low-frequency information of feature maps. This method is straightforward to implement, and introduces no extra hyperparameter. Experimental results on various benchmarks demonstrate the effectiveness of our method. To summarize, our contributions are as follows:

1. We propose a novel domain adaptation technique called LFM. We show that the LFM can help CNNs achieve better generalization performance by utilizing the inherent low-frequency information of feature maps in various domain adaptation tasks.

2. We propose two different ways to utilize the LFM and validate the effectiveness of our method on standard benchmarks for different tasks, such as image classification and object detection.

3. Our method achieves state-of-the-art performance on VisDA-2017 [11] and Cityscapes [12] to FoggyCityscapes [13].

2. APPROACH

In this section, we first introduce the domain adaptation problem and provide a discussion about the characteristics of the low-frequency information. Then, we reveal the relationship between the domain adaptation problem and the low-frequency information. Finally, we analyze our proposed LFM and how to use the LFM.

2.1. Domain Adaptation

The domain adaptation problem can be viewed as aligning part or global feature representations in the learned feature extractor. For example, we consider classification tasks where X is the input space and Y is the set of possible labels. In fact, we have two different distributions over $X \times Y$, called the source and the target domains. An UDA learning algorithm is provided with labeled samples drawn from the source domain, and unlabeled samples drawn from the target domain. The purpose of the UDA algorithm is to make the samples of the same annotation in different domains eventually outputs similar feature representations eventually.

Researchers [1, 2, 4] take the *source-finetune* method as the basic method of domain adaptation. The *source-finetune* makes the CNN model directly train on the source domain data and predict on the target domain data. Taking the classification task with ResNet [16] as an example, at training time, the optimization process is given with Eq (1), where $\hat{\boldsymbol{\theta_f}}, \hat{\boldsymbol{\theta_g}}$ are optimized parameters trained on the source domain $X_S \times Y_S$. The $\boldsymbol{\theta}_g$ represents the parameters of linear classification layer $\boldsymbol{g}(\cdot)$, and the $\boldsymbol{\theta}_f$ is the parameter of CNN encoder $\boldsymbol{f}(\cdot)$. The the global average pooling layer is $\boldsymbol{p}(\cdot)$, and $\boldsymbol{L}(\cdot)$ is the cross entropy loss.

$$\hat{\theta_f}, \hat{\theta_g} = \text{argmin}_{\theta_f, \theta_g} \left[L \left(g \left(p \left(f(X_S, \theta_f), \theta_g \right) \right), Y_S \right) \right] \quad (1)$$

The existing UDA methods adopt various methods to make the parameters $\hat{\boldsymbol{\theta_f}}, \hat{\boldsymbol{\theta_g}}$ adapt to the target domain. However, these methods do not deal with the domain adaptation from the perspective of frequency.

2.2. Low-frequency Information

According to [8, 9, 10], a single natural image or feature map can be decomposed into a low-frequency component that describes the the structure information and a high-frequency component that describes the rapidly changing fine details and noise. We propose an assumption that that the low-frequency components are the key information for cross-domain tasks for the following reasons: 1) Inspired by the idea above, *the low-frequency information of the image or feature map reflects shape information*. Although the overall data distribution is different in different domains, the same objects of the same class label have similar shapes. We hypothesize that the shape information can represent the intrinsic characteristics of the objects. 2) In addition, from the perspective of the signal process, *the low-frequency information represents the main components of a two-dimensional signal* [8]. Utilizing the low-frequency information does not change the main components. 3) Moreover, as shown in Fig. 1, we find that *the domain-discrepancy between the **W** and **A** domain*

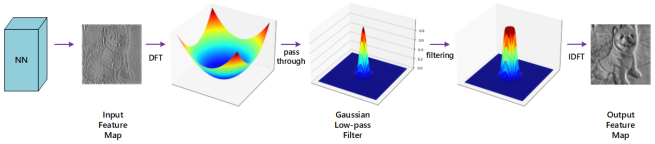

Fig. 2: The processing procedure of LFM. The input spatial feature map obtained by the neural network (NN) is converted to the distribution of frequency by Discrete Fourier Transform (DFT) [9].

is reduced by the digital Gaussian low-pass filter. We conclude that the low-frequency information is more domain-invariant than the whole information. The low-pass filter passes the low-frequency information while suppressing the high-frequency information. Hence, we argue that the suppressed high-frequency information contains domain-related information. Simultaneously, we argue that part of the reason for the domain-discrepancy is the significant differences in high-frequency information between different domains. 4) Notably, the experimental results also demonstrate our assumption in Experiment 3.2.1.

In conclusion, it is reasonable to assume that the low-frequency information is more domain-invariant and suitable for domain adaptation tasks while high-frequency information may harm the generalization performance and stability of the model.

2.3. The Low-Frequency Module for Domain Adaptation

In this section, we propose a low-frequency module (LFM) to help the network align low-frequency feature representations.

2.3.1. The LFM

The LFM is essentially a digital low-pass filter. The principle of the low-pass filter is to pass the low-frequency information while suppressing the high-frequency information for two-dimensional discrete signal [9]. In this paper, the digital low-pass filter adopts the Gaussian low-pass filter [9] with kernel $m \times m$. Because the Gaussian low-pass filter has no ringing [9], it makes the quality of extracted low-frequency information of the whole information better than other low-pass filters, such as the ideal low-pass filter [9].

As shown in Fig. 2, the spatial feature map obtained by neural network (NN) is converted to the distribution of frequency by DFT. Assuming the distribution of input feature map in the frequency domain, the high-frequency information of the output is filtered out when the input passes through the Gaussian low-pass filter. Finally, the output feature map is obtained by Inverse Discrete Fourier Transform (IDFT). Hence, the high-frequency information is suppressed when the value of frequency exceeds the cut-off frequency.

In order to reduce the calculation, we convert the Gaussian low-pass filter from the frequency domain to the spatial domain. The function of the digital spatial Gaussian low-pass filter $\boldsymbol{G}(\cdot)$ is defined as Eq (2), where $-\lfloor m/2 \rfloor \leq x \leq \lfloor m/2 \rfloor, -\lfloor m/2 \rfloor \leq y \leq \lfloor m/2 \rfloor$.

$$G(x,y) = \frac{1}{2\pi \lfloor m/2 \rfloor^2} e^{-(x^2+y^2)/(2\lfloor m/2 \rfloor^2)} \tag{2}$$

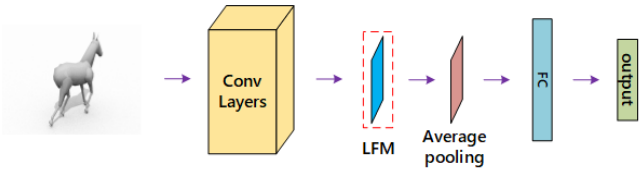

Fig. 3: An overview of the IE domain adaptation model. IE: Insert the end of network. LFM: Low-frequency module.

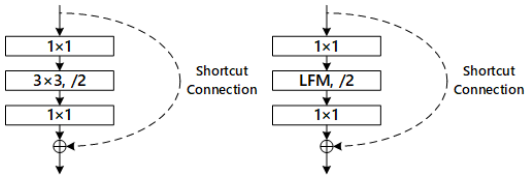

Fig. 4: Visualization of the normal and RSL-equipped bottleneck. **Left** consists of 3×3 strided-convolutions and 1×1 convolutions. **Right** consists of the RSL and 1×1 convolution structure. RSL: Replace strided-convolution layers.

2.3.2. The way to utilize the LFM

The LFM operates on the feature maps to obtain the low-frequency information of feature maps. There are two ways to utilize the LFM in the network:

Insert the end of network (IE). We insert the LFM before the global average pooling layer to extract low-frequency information contained in feature maps as shown in Fig 3. This design can ensure that the feature maps processed by the linear classification layer are the low-frequency information. The IE optimization method of the network is given with Eq (3,4).

$$\hat{\theta_f}, \hat{\theta_g} = \operatorname{argmin} F(\theta_f, \theta_g) \tag{3}$$

$$F(\theta_f, \theta_g) = L\left(g\left(p\left(LFM\left(f\left(X_S, \theta_f\right), \theta_g\right)\right)\right), Y_S\right) \tag{4}$$

Replace strided-convolution layers (RSL). The downsampling operation of CNNs can extract low-frequency information because the operation can result in reducing the size of feature map. Nevertheless, it is unstable and prone to lose crucial information since the operation does not obey the Nyquist Theorem according to [17]. Hence, we replace strided-convolution layers with the LFM in the encoder network as shown in Fig 4. Different from strided-convolution, the LFM performs the low-pass filtering operation on each input feature map and its parameters are fixed.

2.4. Combined with Other Methods

Different from the existing UDA methods, our method deals with the domain adaptation problem from the perspective of

Table 1: Results of the different strategies. The mean accuracy over six tasks on Office-31 is reported based on ResNet-50 [16]. Our methods are trained with Gaussian high-pass pre-processing images, Gaussian low-pass pre-processing images, insert the end of network and replace strided-convolution layers, respectively.

Dataset	High-pass Pre-process	Low-pass Pre-process	IE	RSL	Average
Office-31					76.1
	✓				73.2
		✓			78.0
			✓		81.4
				✓	**81.6**

frequency. Hence, our method is different from other methods in dealing with problems. Our method is formulated as a plug-and-play unit that can be used to combine with existing UDA methods to achieve better generalization performance. In the Experiments, we apply [4], [18], and [19] to assist our method, and achieve state-of-the-art performance on multiple computer vision tasks.

3. EXPERIMENTS

3.1. Experimental Setups

The 3×3 convolution is the current popular structure. Hence, the m of our LFM sets as 3 by default. To show the effectiveness of the proposed LFM, we first perform small image classification experiments for domain adaptation on the Office-31 [15] dataset to verify our method. On Office-31, similar to [3, 1], we validate the pairwise domain adaptation performance of our method on all six pairs of domains and take the average accuracy. Then we experiment with a challenging test-bed for UDA with the domain shift from synthetic data to real imagery on VisDA-2017 [11]. On VisDA-2017, we follow the full protocol [4] for the training setting but D_0 is set as 0.85, unlike the original 1.0. Because D_0 represents the cluster limit threshold, our method brings the same classes closer, and the threshold setting should be stricter. To explore LFM's generality further, we also conduct multi-label object detection experiments from Cityscapes [12] to FoggyCityscapes [13], and we follow these two settings [19] and [18] and finetune the network for adaptation experiments from Cityscapes to FoggyCityscapes. All models are trained from scratch on NVIDIA V100 GPUs with the default data augmentation and training strategy which are optimized for the vanilla model and no other tricks are used.

3.2. Ablation Studies

3.2.1. Effect of the different frequency components

The *Source-finetune* is the baseline method for cross-domain task. Hence, we first test the result of source-finetune on Office-31 dataset. As shown in the Table 1, the first line shows the result of baseline method (76.1).

To validate the effect of high-frequency information for cross-domain problem, we adopt Gaussian high-pass filtering to pre-process the Office-31 datasets. The result reveals that the high-frequency information of images limits the generalization ability of the model (from 76.1 to 73.2) in the second line. It is reasonable that the high-frequency information of image data contains domain-related information. Furthermore, we utilize Gaussian low-pass filtering to pre-process the Office-31 datasets to verify the effectiveness of low-frequency information. It can be observed that the result is better than the source-finetune (from 76.1 to 78.0), which means the low-frequency information is beneficial to alleviate domain adaptation task. From above results, it proves that our assumption that the low-frequency information is more domain-invariant while the high-frequency information contains domain-related information.

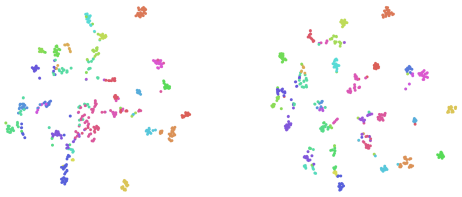

Fig. 5: Visualization with t-SNE for different methods. **Left**: t-SNE of *source-finetune*. **Right**: IE. The input activations of the last fully-connected layer are used for the computation of t-SNE. The results are Office-31 task **A** $\rightarrow$ **D**. The same color represents the same class while different color means different category.

3.2.2. Effect of the LFM method

It is noted that we adopt the IE strategy to train models to further utilize the low-frequency information of the dataset, as shown in Fig 1. It can be observed that introducing the IE further improves the adaptation performance and is better than the operation of Gaussian low-pass filtering (from 78.0 to 81.4). This phenomenon shows that it is better for the network to adaptively extract low-frequency features than the pre-processing dataset. Finally, we also adopt the RSL strategy to train models, and the result show impressive performance and are sightly better than IE (from 81.4 to 81.6). The two results demonstrate the effectiveness of our designs. Meanwhile, we visualize the distribution of learned features by t-SNE. As shown in Fig. 5, it illustrates a representative task **A** $\rightarrow$ **D**. Compared to source-finetune, the target feature representations learned by IE demonstrate higher intra-class compactness and a much larger inter-class margin. This suggests that utilizing low-frequency information can extract features that are invariant to different domains.

3.2.3. Compared with other methods

The LFM method is compared with two existing basic mainstream methods in the UDA field: RevGrad [3], and DAN [1], to verify the merit of the proposed LFM. As Table 2 shows, both the IE and RSL methods are better than the DAN method, and the RevGrad method is slightly better than the IE and RSL methods. It should be emphasized that our method alleviates the cross-domain problem from the perspective of frequency, and it is different from the existing methods. Therefore, our method is orthogonal and complementary to the existing methods. For verifying the conjunction of our LFM method, we adopt the DAN and RevGrad to assist our method, respectively.

Firstly, we utilize the DAN method to assist the IE and the final result is better than the DAN (from 80.4 to 82.3). Then, we also construct the RSL experiment and its result is also better than that of DAN (from 80.4 to 82.3). The performance of *RSL+DAN* is equivalent to that of *IE+DAN*. Although the performance of IE and RSL is better than that of DAN, the performance of the combination of the LFM method and DAN outperforms the single method.

For the RevGrad method, we apply it to assist the IE method and its performance is better than that of RevGrad (from 82.2 to 83.1). Similarly, the result of RSL is also better than the RevGrad (from 82.2 to 83.2). Meanwhile, the result of *RSL+RevGrad* is slightly better than that of *IE+RevGrad* (from 83.1 to 83.2).

These results show the effect of the LFM method for alleviating the domain adaptation problem and prove our low-frequency assumption.

3.3. Comparison with the State-of-the-art

For fair comparison, we adopt the same backbone and re-implement them. Our re-implementations achieve comparable performance compared to original papers.

3.3.1. Classification results

VisDA is a challenging testbed for UDA with the domain shift from synthetic data to real imagery. In total there are ∼280k images from 12 categories. The images are split into three sets, a training set with 152,397 synthetic images, a validation set with 55,388 real-world images, and a test set with 72,372 real-world images. As shown in Table 3, the *Average* indicates the classification accuracy of 12 classes on VisDA-2017 with the validation set as the target domain by utilizing different UDA methods. Our method outperforms the popular UDA methods: RevGrad, DAN, self-ensembling (SE) (the first place in VisDA-2017 competition), and CAN. The mean accuracy of our RSL method outperforms that of the current state-of-the-art method CAN by 0.5 (from 86.8 to 87.3) on the VisDA-2017 validation dataset and the *IE+CAN* method is slightly better than the *RSL+CAN* method (from 87.3 to 87.4). On such a large dataset, the results reveal the potential and effectiveness of LFM.

Table 2: Classification accuracy (%) for all the six tasks of Office-31 dataset based on ResNet-50 [16].

Method	A → W	D → W	W → D	A → D	D → A	W → A	Average
Source-finetune	68.4 ± 0.2	96.7 ± 0.1	99.3 ± 0.1	68.9 ± 0.2	62.5 ± 0.3	60.7 ± 0.3	76.1
DAN [1]	80.5 ± 0.4	97.1 ± 0.2	99.6 ± 0.1	78.6 ± 0.2	63.6 ± 0.3	62.8 ± 0.2	80.4
RevGrad [3]	82.0 ± 0.4	96.9 ± 0.2	99.1 ± 0.1	79.7 ± 0.4	68.2 ± 0.4	67.4 ± 0.5	82.2
Ours (IE+Source-finetune)	77.3 ± 0.2	96.7 ± 0.2	**99.8 ± 0.2**	83.0 ± 0.2	65.8 ± 0.2	65.6 ± 0.2	81.4
Ours (RSL+Source-finetune)	77.5 ± 0.2	97.0 ± 0.2	**99.8 ± 0.2**	83.2 ± 0.2	66.2 ± 0.2	66.0 ± 0.2	81.6
Ours (IE+DAN)	80.3 ± 0.2	97.0 ± 0.2	**99.8 ± 0.2**	**83.4 ± 0.2**	66.8 ± 0.2	66.3 ± 0.2	82.3
Ours (RSL+DAN)	80.4 ± 0.2	97.1 ± 0.2	**99.8 ± 0.2**	83.2 ± 0.2	67.0 ± 0.2	66.0 ± 0.2	82.3
Ours (IE+RevGrad)	**82.6 ± 0.3**	96.9 ± 0.2	**99.8 ± 0.2**	82.8 ± 0.4	68.8 ± 0.3	**68.0 ± 0.4**	83.1
Ours (RSL+RevGrad)	82.5 ± 0.2	**97.3 ± 0.2**	**99.8 ± 0.2**	83.1 ± 0.4	**69.1 ± 0.3**	67.5 ± 0.4	**83.2**

Table 3: Classification accuracy (%) on the VisDA-2017 validation set based on ResNet-101 [16].

Method	airplane	bicycle	bus	car	horse	knife	motorcycle	person	plant	skateboard	train	truck	Average
Source-finetune	72.3	6.1	63.4	**91.7**	52.7	7.9	80.1	5.6	90.1	18.5	78.1	25.9	49.4
RevGrad [3]	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
DAN [1]	68.1	15.4	76.5	87.0	71.1	48.9	82.3	51.5	88.7	33.2	**88.9**	42.2	62.8
JAN [2]	75.7	18.7	82.3	86.3	70.2	56.9	80.5	53.8	92.5	32.2	84.5	54.5	65.7
GSDA [20]	93.1	67.8	83.1	83.4	94.7	93.4	93.4	79.5	93.0	88.8	83.4	36.7	81.5
SE [21]	95.9	87.4	85.2	58.6	96.2	95.7	90.6	80.0	94.8	90.8	88.4	47.9	84.3
CAN [4]	96.7	**90.3**	84.2	66.4	96.5	97.1	88.0	83.0	96.1	95.0	87.0	**61.3**	86.8
Ours (RSL+CAN)	**97.5**	86.1	84.7	71.7	**96.2**	**98.2**	90.6	82.7	96.8	94.8	**88.9**	59.5	87.3
Ours (IE+CAN)	96.8	85.8	**85.3**	72.8	95.8	97.3	**91.7**	**84.0**	**97.3**	**95.1**	87.1	59.8	**87.4**

3.3.2. Object detection results

To verify the generality of our method, we construct object detection experiments, and adopt current state-of-the-art methods: DA-Faster-ICR-CCR [19] and KR-DA-Faster [18] as our baseline methods.

First, we train our method with the state-of-the-art method KR-DA-Faster. Its backbone is widely popular ResNet-50 and the network initializes with Caffe pre-trained weights. Ultimately, the model achieves the best performance thus far from Cityscapes to Foggy-Cityscapes by adopting our method. Table 4 shows the comparison results. Our *IE+KR* can boost the performance of KR-DA-Faster by 0.6 mAP (from 40.8 to 41.4). The *RSL+KR* method adopts Pytorch pre-trained weights and still outperforms the KR-DA-Faster by 1.3 mAP (from 40.8 to 42.1) although Caffe pre-trained models have better performance than Pytorch pre-trained. In particular, our RSL can greatly improve the detection results in the target domain. The results reveal that the RSL is better than IE in the object detection task. In particular, our method can greatly improve the detection results for some difficult categories such as "train". The RSL method outperforms the state-of-the-art by 5.1 mAP for the training class. This clearly verifies the importance of low-frequency information for cross-domain object detection.

Moreover, DA-Faster-ICR-CCR is an extension method based on DA-Faster. Its backbone is VGG-16. We combine our RSL idea with VGG-16 and apply it to the DA-Faster-ICR-CCR method. As shown in Table 5, we observe that our method outperforms DA-Faster-ICR-CCR by 2.0 mAP (from 29.9 to 31.9). The result demonstrates that our method is

Table 4: Results (%) on adaptation from Cityscapes to Foggy-Cityscapes (normal → foggy). The backbone network is ResNet-50.

Methods	person	rider	car	truck	bus	train	motorcycle	bicycle	mAP
Source-only	26.9	38.2	35.6	18.3	32.4	9.6	25.8	28.6	26.9
SC-DA-Faster [6]	33.8	42.1	52.1	26.8	42.5	26.5	29.2	34.5	35.9
GPA [22]	32.9	46.7	54.1	24.7	45.7	41.1	32.4	38.7	39.5
KR-DA-Faster [18]	36.8	46.4	54.5	27.7	47.3	42.7	32.7	38.6	40.8
Our (IE+KR)	36.8	46.9	52.9	**28.9**	48.2	47.1	31.7	38.9	41.4
Our (RSL+KR)	**37.1**	**47.6**	**55.0**	28.3	**48.5**	**47.8**	**32.8**	**39.8**	**42.1**

Table 5: Results (%) on adaptation from Cityscapes to Foggy-Cityscapes (normal → foggy). The backbone network is VGG-16.

Method	persn	rider	car	truck	bus	train	mbike	bcycle	mAP
Source Only	24.1	33.1	34.3	4.1	22.3	3.0	15.3	26.5	20.3
DA-Faster [5]	25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
DA-Faster-ICR-CCR [19]	29.7	37.3	**43.6**	20.8	**37.3**	12.8	**25.7**	31.7	29.9
Our method (RSL+DA-Faster-ICR-CCR)	**30.1**	**42.9**	43.3	**24.1**	35.2	**20.5**	25.4	**33.6**	**31.9**

compatible with other backbones.

4. CONCLUSION

In this paper, we propose an assumption that low-frequency information is more domain-invariant and more suitable for domain adaptation tasks while the high-frequency information contains domain-related information in different domains. Meanwhile, we construct massive experiments and visualization analysis to demonstrate the assumption. Finally, we introduce a method, named LFM, to combine with existing UDA methods easily and achieve better performance. Our method outperforms state-of-the-art methods on VisDA-2017 and Cityscapes to FoggyCityscapes. In future, we will introduce RSL to the current self-supervised methods [23, 24] we have already explored.

Acknowledgement. This work was supported by Key-Area Research and Development Program of Guangdong Province (No.2021B0101410003), National Natural Science Foundation of China under Grants No.62002357, No.62176254, No.61976210, No.61876086, No.62076235 and No.62006230.

5. REFERENCES

[1] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan, "Learning transferable features with deep adaptation networks," in *ICML*, 2015, pp. 97–105.

[2] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan, "Deep transfer learning with joint adaptation networks," in *ICML*, 2017, pp. 2208–2217.

[3] Yaroslav Ganin and Victor Lempitsky, "Unsupervised domain adaptation by backpropagation," in *ICML*, 2015, pp. 1180–1189.

[4] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *CVPR*, 2019, pp. 4893–4902.

[5] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *CVPR*, 2018, pp. 3339–3348.

[6] Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin, "Adapting object detectors via selective cross-domain alignment," in *CVPR*, 2019, pp. 687–696.

[7] Qi Cai, Yingwei Pan, Chong-Wah Ngo, Xinmei Tian, Lingyu Duan, and Ting Yao, "Exploring object relation in mean teacher for cross-domain detection," in *CVPR*, 2019, pp. 11457–11466.

[8] Russell L DeValois and Karen K DeValois, *Spatial vision*, vol. 14, Oxford university press, 1990.

[9] Rafael C. Gonzalez and Richard E. Woods, "Digital image processing," *Prentice Hall International*, vol. 28, no. 4, pp. 484 – 486, 2008.

[10] Yunpeng Chen, Haoqi Fan, et al., "Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution," in *ICCV*, 2019, pp. 3435–3444.

[11] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko, "Visda: The visual domain adaptation challenge," *arXiv*, 2017.

[12] Marius Cordts, Mohamed Omran, et al., "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016, pp. 3213–3223.

[13] Christos Sakaridis, Dengxin Dai, and Luc Van Gool, "Semantic foggy scene understanding with synthetic data," *IJCV*, vol. 126, no. 9, pp. 973–992, 2018.

[14] Van Der Maaten Laurens and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[15] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell, "Adapting visual category models to new domains," in *ECCV*, 2010, pp. 213–226.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[17] Richard Zhang, "Making convolutional networks shift-invariant again," in *ICML*, 2019, pp. 7324–7334.

[18] Krumo, "Domain-Adaptive-Faster-RCNN-PyTorch," in *https://github.com/krumo/Domain-Adaptive-Faster-RCNN-PyTorch*, 2019.

[19] Chang-Dong Xu, Xing-Ran Zhao, Xin Jin, and Xiu-Shen Wei, "Exploring categorical regularization for domain adaptive object detection," in *CVPR*, 2020, pp. 11724–11733.

[20] Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen, "Unsupervised domain adaptation with hierarchical gradient synchronization," in *CVPR*, 2020, pp. 4043–4052.

[21] Geoffrey French, Michal Mackiewicz, and Mark H. Fisher, "Self-ensembling for visual domain adaptation," in *ICLR*, 2018.

[22] Minghao Xu, Hang Wang, Bingbing Ni, Qi Tian, and Wenjun Zhang, "Cross-domain detection via graph-induced prototype alignment," in *CVPR*, 2020, pp. 12355–12364.

[23] Zhaowen Li, Zhiyang Chen, et al., "Mst: Masked self-supervised transformer for visual representation," *NeurIPS*, vol. 34, 2021.

[24] Zhaowen Li, Yousong Zhu, Rui Zhao, et al., "Univip: A unified framework for self-supervised visual pre-training," in *CVPR*, 2022.