applied sciences

MDPI

Article

P-Norm Attention Deep CORAL: Extending Correlation Alignment Using Attention and the P-Norm Loss Function

Zhi-Yong Wang [1] and Dae-Ki Kang [2,*]

[1] Department of Computer Software, Weifang University of Science and Technology, Shouguang 262700, China; wangzhiyong@wfust.edu.cn
[2] Department of Computer Engineering, Dongseo University, Busan 47011, Korea
* Correspondence: dkkang@dongseo.ac.kr; Tel.: +82-51-320-1724

Abstract: CORrelation ALignment (CORAL) is an unsupervised domain adaptation method that uses a linear transformation to align the covariances of source and target domains. Deep CORAL extends CORAL with a nonlinear transformation using a deep neural network and adds CORAL loss as a part of the total loss to align the covariances of source and target domains. However, there are still two problems to be solved in Deep CORAL: features extracted from AlexNet are not always a good representation of the original data, as well as joint training combined with both the classification and CORAL loss may not be efficient enough to align the distribution of the source and target domain. In this paper, we proposed two strategies: attention to improve the quality of feature maps and the p-norm loss function to align the distribution of the source and target features, further reducing the offset caused by the classification loss function. Experiments on the Office-31 dataset indicate that our proposed methodologies improved Deep CORAL in terms of performance.

Keywords: attention; Deep CORAL; domain adaptation

Citation: Wang, Z.-Y.; Kang, D.-K. P-Norm Attention Deep CORAL: Extending Correlation Alignment Using Attention and theP-Norm Loss Function. *Appl. Sci.* **2021**, *11*, 5267. https://doi.org/10.3390/app11115267

Academic Editors: Wonjoon Kim, Sekyoung Youm and Sungbum Jun

Received: 4 May 2021
Accepted: 2 June 2021
Published: 6 June 2021

1. Introduction

Deep learning-based applications have outperformed the imagination of human beings in many aspects, such as computer vision, speech recognition, natural language processing, audio recognition, etc. [1–3], but domain shifts dramatically damage the performance of deep learning methods [4,5]. In such a scenario, features extracted by a deep neural network, which was pre-trained using existing datasets (called the source domain), can become meaningless for the target task (referred to as the target domain). Essentially, the different data distributions between the source and target domain will hinder the generalization on the target task, which means the learned knowledge from source domains cannot be transferred to target domains.

To relieve the domain shift issue, which is common in practical scenarios, collecting labeled data and training a new classifier for every possible scenario can compensate the degradation in performance. However, the cost of acquiring huge volumes of labeled data remains expensive and time consuming. Domain Adaptation (DA) [6] is an alternative solution, which, instead of collecting labeled data, utilizes known or labeled data to learn a classifier for unknown or unlabeled data. Domain adaptation is a particular case of Transfer Learning (TL), which has become commonplace in today's deep learning-centric computer vision.

2. Related Work

CORrelation ALignment (CORAL) [7] works well by aligning the distribution of the source and target features in an unsupervised manner. However, it only relies on a linear transformation to minimize the squared Frobenius norm distance of the covariances of the source and target features, which will limit flexibility and adaptability. Furthermore, CORAL needs to calculate the second-order statistics (covariances) at first between the

source and target data and, after that, transform the source domain to the target domain to align their distributions. Training an extra classifier, such as Support Vector Machine (SVM), is necessary with transformed source domain data and, finally, classifying the target domain dataset. In this slightly tedious process, an external classifier must be involved to obtain the final category, which we call "not-end-to-end".

Deep CORAL [8] has been proposed using Deep Neural Networks (DNNs), which is a kind of nonlinear transformation to extend CORAL. Deep CORAL adds the objective function of basic CORAL to be a part of the total loss function, making full use of the characteristics of DNNs, which can minimize the loss function to align covariances between the source and target domain. Hence, Deep CORAL essentially overcomes the linear transformation dependence of CORAL attributed to the nonlinear characteristics of DNNs. Meanwhile, in order to address the not-end-to-end dilemma, Deep CORAL introduces joint training into neural network to reduce the influence of degenerated features induced by minimizing the CORAL loss alone. Nevertheless, we can still point out several problems existing in Deep CORAL.

First of all, Deep CORAL is not concerned with the quality of data, which will influence the accuracy. Deep CORAL extracts features of the source and target datasets using AlexNet only. AlexNet [9], designed primarily by Alex Krizhevsky, is a Convolutional Neural Network (CNN), which became famous in 2012 since the championship with an error rate of 15.3% in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC-2012). However, not all features extracted from convolutional layers can perfectly represent the original data. Our experiments illustrated this point (see Section 4).

Secondly, according to [8], the AlexNet used by Deep CORAL could project the source and target domain to a single point because Deep CORAL relies on the CORAL loss only. Therefore, joint training with both the classification loss and CORAL loss has been chosen by Deep CORAL to reduce this situation, but the classification loss could result in an offset when Deep CORAL tries to align the distributions of the source and target domain with minimizing CORAL loss.

Basically, minimizing CORAL loss only may align the second-order statistics of the source and target domain properly. However, if classification loss is added to the CORAL loss, then it will be a redundant term for CORAL, so the alignment will be disturbed because of it. To overcome this problem, we imported the p-norm [10] to further align the distributions and improve the generalization accuracy.

In this paper, we introduced P-norm Attention Deep CORAL (P-ADC) to address the above challenges. The key insight underlying P-ADC is that we added attention into DNN of Deep CORAL, which not only retained the advantages of AlexNet, but also considered the use of attention to highlight image features, which had the effect of image preprocessing. Meanwhile, our experimental results show that Attention Deep CORAL provided an effective improvement when compared with traditional Deep CORAL. Furthermore, we extended the loss function of Deep CORAL, which included two parts into $n \in [1, \infty)$ parts, to ease the second challenge mentioned above. The first part of the extended loss function still maintained the original classification loss function, and the rest we introduced contained the p-norm to balance the offset caused by the classification loss.

3. Method

Suppose the source domain training set $D_S = \{x_i, y_j\}, x \in \mathbb{R}^d, i \in \{1, \cdots, n_S\}, j \in \{1, \cdots, L\}$, consists of N image-label pairs (x_i, y_j) where x_i is a source domain image, while y_j is its corresponding label, and the target domain data $D_T = \{u_i\}, u \in \mathbb{R}^d$, which are unlabeled. In the meantime, n_S, n_T, μ_s, μ_t, and C_S, C_T are the number, the feature vector means, and covariance matrices of the source and target data, respectively.

3.1. CORrelation ALignment

CORAL works by aligning the distributions of the source and target features in an unsupervised manner, matching the distributions by aligning the second-order statistics

and the covariance, and applying a linear transformation M to minimize the Frobenius distance metric.

$$\min_M \|C_{\hat{S}} - C_T\|_F^2 = \min_M \|M' C_S M - C_T\|_F^2 \quad (1)$$

where $C_{\hat{S}}$ is the covariance of the transformed source features and C_S and C_T are the covariance matrices. Let $C_S = U_S \Sigma_S U_S^{'}, C_T = U_T \Sigma_T U_T^{'}$ be the singular-value decomposition (SVD) of C_S, C_T. Then, the final optimal value of M as $M^* = \left(U_S \Sigma_S^{+\frac{1}{2}} U_S^{'}\right) \left(U_{T[1:r]} \Sigma_{T[1:r]}^{\frac{1}{2}} U_{T[1:r]}^{'}\right)$ where Σ^+ denotes the Moore–Penrose pseudoinverse of Σ.

3.2. Deep CORAL

Deep CORAL minimizes the difference in the covariance between the source and target domain with the aid of a DNN. We defined the CORAL loss (Equation (2)) as a part of the total loss function (Equation (3)). Figure 1 shows the architecture of Deep CORAL.

$$l_{CORAL} = \frac{1}{4d^2} \|C_S - C_T\|_F^2 \quad (2)$$

$$l_{TOTAL} = l_{CLASS} + l_{CORAL} = l_{CLASS} + \sum_{i=1}^{n} \lambda_i \frac{1}{4d^2} \|C_S - C_T\|_F^2 \quad (3)$$

where l_{CLASS} indicates the classification loss function, e.g., cross-entropy, square-loss, etc. Cross-Entropy was adopted in our experiments.

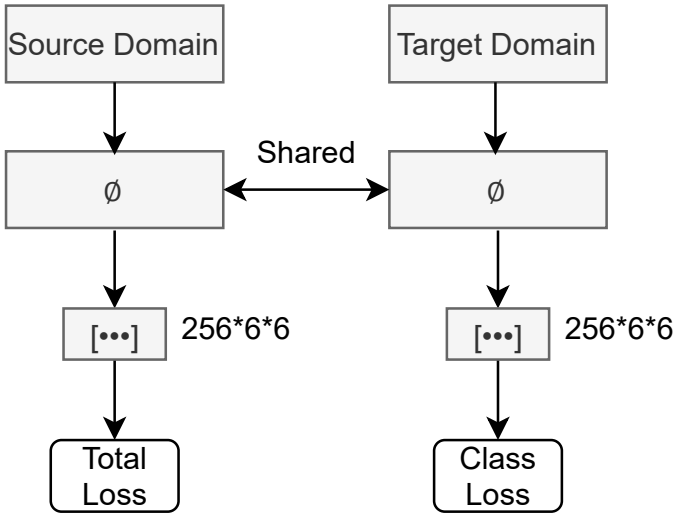


Figure 1. The architecture of Deep CORAL. ϕ denotes any deep neural network (e.g., AlexNet). The 256 * 6 * 6 denotes the size of the features maps extracted by AlexNet; the 256 stands for the number of channels, and 6 * 6 is the weight * height of a single feature map.

3.3. Our Method

Deep CORAL model was built using AlexNet in which convolutional layers are inefficient for modeling global dependencies in images due to its local view. We adapted the attention mechanism to overcome the shortcoming of AlexNet, enabling the image features extracted by convolutional layers to be able to provide more representative information. We call the proposed method P-norm Attention Deep CORAL (P-ADC) because of the added attention mechanism (see Figure 2).

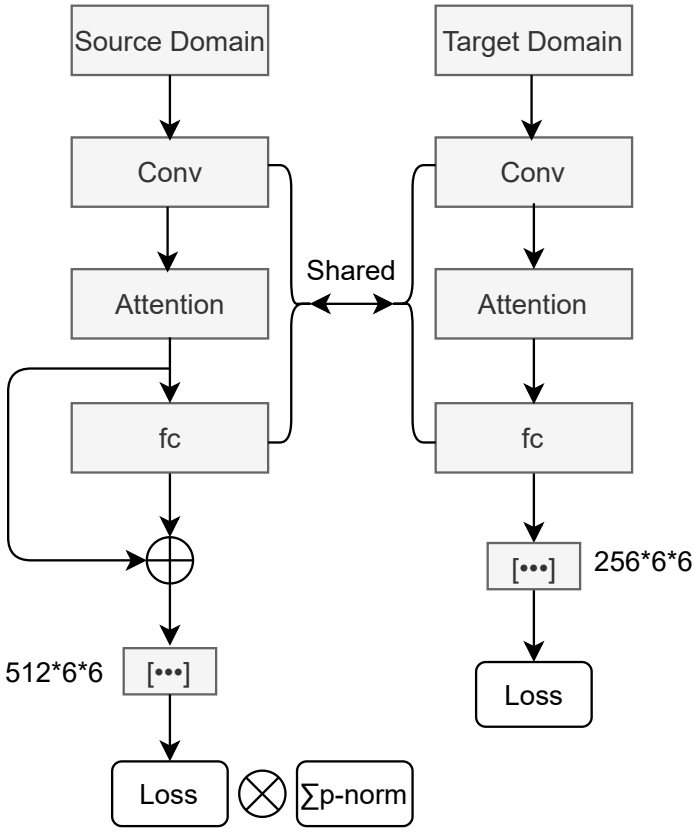


Figure 2. The architecture of P-ADC. $\oplus$ denotes the stack operation. For example, if the size of a and b is $256 \times 6 \times 6$, respectively, then the size of $a \oplus b$ should be $512 \times 6 \times 6$. $\otimes$ is the addition operation, i.e., classification loss + p-norm loss.

Suppose image features $X = \{x_1, x_2, \cdots, x_N\}$ are provided from the previous layer where $x_i \in \mathbb{R}^{C \times W \times H}$ are the image features of each sample. Note that C denotes the number of channels, and W, H is the width and height of the image features x. Then, the energy of x_i is expressed as follows:

$$s^i_{j,k} = \sum_{l=1}^{C} v^i_{j,l} t^i_{l,k}, v^i = (W_V x_i)', t^i = \tanh\big((W_K x_i) + (W_Q x_i)\big) \tag{4}$$

where $W_V \in \mathbb{R}^{C \times C}$,$W_K \in \mathbb{R}^{C \times C}$,$W_Q \in \mathbb{R}^{C \times C}$ are the learned weight matrices, which belong to the convolutional layer with kernel_size = 1, stride = 1, and padding = 0. C is the number of channels. Here, we can reduce the channel number C to $\frac{C}{k}$ [11], where $k = 8$ was chosen in our experiment to reduce the number of parameters while not decreasing the performance significantly. v^i and t^i are two different feature spaces calculated with the image feature map x_i of the previous hidden layer. s^i means the energy of x_i. j and k indicate the position coordinates of the energy s^i of the ith image sample.

Attention mechanisms [12,13] have been employed successfully in sequence modeling and transduction problems such as speech recognition, neural captioning, etc., to tackle capturing long-range interactions for convolutions. Recently, attention mechanisms have also been applied in computer vision models to provide contextual information. The essence of the attention mechanism is actually an addressing process: Given a query vector

q related to the task and a key vector k, the distribution values will be calculated by q and k, and then, attach it to the value vector v. The main attention models are as follows.

$$\text{Additive model} \quad s(q) = v^T tanh(Wk + Uq) \tag{5}$$

$$\text{Dot-product model} \quad s(q) = k^T q \tag{6}$$

$$\text{Scaled dot-product model} \quad s(q) = \frac{k^T q}{\sqrt{d}} \tag{7}$$

$$\text{Bilinear model} \quad s(q) = k^T W q \tag{8}$$

Equation (4) belongs to a kind of additive model.

The attention matrices of x_i are given by:

$$\alpha_{j,k}^i = \frac{e^{\left(s_{j,k}^i\right)}}{\sum_{k=1}^{M} e^{\left(s_{j,k}^i\right)}} \tag{9}$$

where $M = W \times H$ and α^i denote the attention matrices of x_i.

Then, the output of the attention layer is:

$$o_{j,k}^i = \sum_{l=1}^{M} v_{j,l}^i \alpha_{k,l}^i \tag{10}$$

where $o^i \in \mathbb{R}^{C \times W \times H}$ is the output of the attention layer. According to [13], additive attention and dot-product attention are the most popular attention functions. Here, we also defined dot-product attention for the convenience of application. $s_{j,k}^i = \sum_{l=1}^{C} v_{j,l}^i t_{l,k}^i, v^i = \left(W_V x_i\right)', t^i = W_K x_i$.

The definition of the p-norm is as below:

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}} = \left(|x_1|^p + |x_2|^p + \cdots + |x_n|^p\right)^{\frac{1}{p}}, 1 \le p < \infty \tag{11}$$

We defined the p-norm loss between two domains for a single feature layer.

$$l_{p-norm} = \left(\frac{\|C_S - C_T\|_p}{2d}\right)^p, 1 \le p < \infty \tag{12}$$

where C_S and C_T denote the feature covariance matrices. d was set to the number of categories, i.e., the dimension or the output of the last fully connected layer. Therefore, according to the definition of CORAL loss, we have $l_{CORAL} = l_{2-norm}$. The total loss function is as follows:

$$l_{TOTAL} = l_{CLASS} + \sum_{p} \sum_{i} \lambda_i l_{p-norm}, 1 \le p < \infty \tag{13}$$

where λ trades off the adaptation and classification accuracy on the source domain.

4. Experiment Results

To evaluate our method, we performed experiments on a famous domain adaptation benchmark dataset, the Office-31 dataset [14]. This dataset contains three image domains: DSLR, Amazon, and Webcam, and each of them has 31 classes with corresponding class names.

In Figure 3, we compare the information quantity of feature maps for training with vs. without attention in Amazon. We can clearly see that adding attention helped the classifier acquire much more information, which means we can obtain higher test accuracy after adding attention.

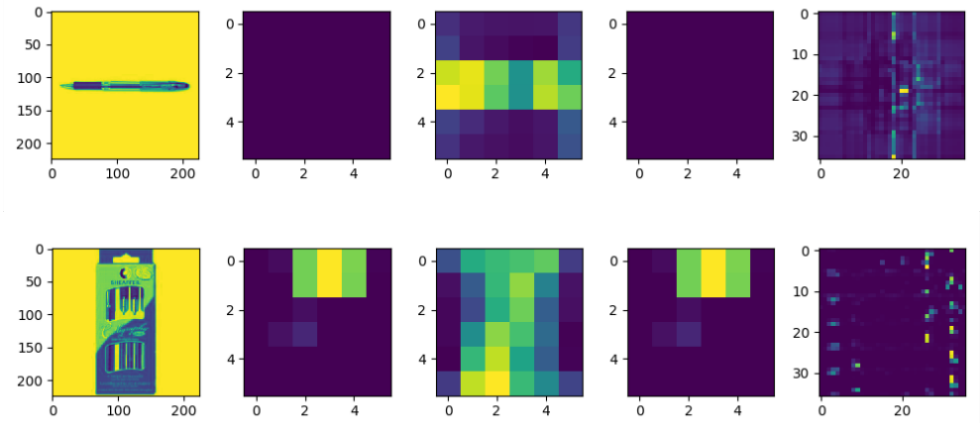


Figure 3. Comparison of feature maps with vs. without attention in Amazon. The 1st column is the original image (sum of 3 channels); the 2nd column is with attention (0 channels); the 3rd column is with attention (sum of 256 channels); the 4th column is without attention (sum of 256 channels); the 5th column is an attention matrix. We can see the amount of information of the 4th column is very small, just as the single channel (i.e., 0 channels) with attention in the 2nd column.

In this experiment, we took one domain as the source domain and another as the target domain. We defaulted to the labels of all source domain data being given, and the labels of all target data were unknown. Because there are three domains in the Office-31 dataset, we can conduct our experiment on six experiment settings, namely: $A \rightarrow W$: (A)mazonas the source domain and (W)ebcam as the target domain; $A \rightarrow D$: (A)mazon as the source domain and (D)SLRas the target domain; $W \rightarrow A$: (W)ebcam as the source domain and (A)mazon as the target domain; $W \rightarrow D$: (W)ebcam as the source domain and (D)SLR as the target domain; $D \rightarrow A$: (D)SLR as the source domain and (A)mazon as the target domain; and $D \rightarrow W$: (D)SLR as the source domain and (W)ebcam as the target domain.

For comparative analysis of our method (P-ADC), in addition to Deep CORAL, we tested other well-known algorithms (deep domain confusion and conditional domain adversarial networks) on the Office-31 benchmark dataset. Deep Domain Confusion (DDC) [15] adds an adaptation layer and domain confusion loss in AlexNet. Conditional Domain Adversarial Networks (CDANs) [16] introduce multilinear conditioning and entropy conditioning to improve the discriminability and guarantee the transferability.

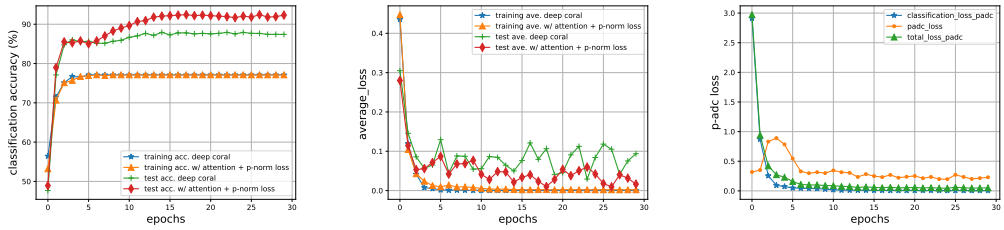
Following [8], we initialized the weight of the last fully connected layer (fc8) with $\mathcal{N}(0.0, 0.005)$ and set the dimension to 31, the number of categories. The other layers of AlexNet were initialized with the pre-trained model parameters of ImageNet [17], keeping the layerwise parameter settings. We also set batch size = 128, learning rate = 10^{-3}, weight decay = 5×10^{-4}, and momentum = 0.9 for all of the experiments below (Table 1) for a fair comparison.

From Table 1, we can see that P-ADC achieved higher average performance than Deep CORAL and the other baseline methods. In three out of six shifts, P-ADC$_{(2-3)}$ achieved the highest accuracy ($l_{TOTAL} = L_C + \sum_{p=2}^{3} \sum_{i=1}^{t} \lambda_i l_{p-norm}$, where t is the number of p-norm loss layers in a deep network and P-ADC$_{(2-3)}$ means that p ranges from two to three). For the other three shifts, P-ADC$_{(2-4)}$ ($l_{TOTAL} = L_C + \sum_{p=2}^{4} \sum_{i=1}^{t} \lambda_i l_{p-norm}$, where P-ADC$_{(2-4)}$ indicates that p ranges from two to four) obtained the best scores. In this experiment, we only tried P-ADC$_{(2-3)}$ and P-ADC$_{(2-4)}$ because the p-norm loss would take up many computing resources with the increase of p, resulting in the computing speed declining dramatically. In addition, as we can see from Table 1, the test accuracy could not achieve the result of the official algorithm for all due to the fine-tuned AlexNet model from PyTorch, as well as the software and hardware environment.

Table 1. Target accuracies for all six domain shifts in the Office-31 dataset (training epoch=100). Note that $L_{(a-b)} = \sum_{p=a}^{b}\sum_{i=1}^{t}\lambda_i l_{p-norm}$, $L_C = l_{CLASS}$, and P-ADC$_{(a-b)}$ = P-ADC$\left(l_{TOTAL} = L_C + L_{(a-b)}\right)$, where l_{CLASS} is the classification loss function, $L_{(a-b)}$ denotes the p-norm loss function where p ranges from a to b, and a and b are natural numbers greater than 1. Bold denotes the highest accuracy.

Method	$A \rightarrow W$	$A \rightarrow D$	$W \rightarrow A$	$W \rightarrow D$	$D \rightarrow A$	$D \rightarrow W$
No Adaptation	45.3 ± 0.7	23.4 ± 1.6	38.0 ± 0.2	47.4 ± 1.6	23.0 ± 0.4	59.1 ± 0.9
DDC	44.3 ± 0.7	24.0 ± 1.5	38.6 ± 0.5	44.6 ± 1.0	24.0 ± 0.7	62.5 ± 0.5
CDAN	42.9 ± 0.9	35.2 ± 1.8	34.8 ± 2.4	73.3 ± 0.7	37.1 ± 0.4	83.3 ± 1.3
CDAN+E	43.9 ± 0.9	36.4 ± 2.4	36.2 ± 0.7	71.6 ± 1.3	34.4 ± 0.5	78.1 ± 1.9
Deep CORAL	54.77 ± 0.68	44.14 ± 1.97	40.19 ± 0.11	74.74 ± 0.44	40.9 ± 0.12	87.62 ± 0.58
Deep CORAL ($l = L_C + L_4$)	51.74 ± 0.55	41.26 ± 1.90	41.03 ± 0.06	75.22 ± 0.52	39.25 ± 0.2	90.69 ± 0.25
Deep CORAL+Att	55.13 ± 0.29	48.16 ± 1.29	41.72 ± 0.21	74.88 ± 0.58	40.43 ± 0.11	89.26 ± 0.20
P-ADC$_{(2-3)}$	55.45 ± 0.24	**49.32 ± 1.34**	**41.89 ± 0.24**	**75.70 ± 0.40**	41.82 ± 0.07	91.38 ± 0.20
P-ADC$_{(2-4)}$	**56.88 ± 0.67**	49.02 ± 1.05	41.01 ± 0.21	74.62 ± 0.52	**41.93 ± 0.30**	**91.45 ± 0.28**

Figure 4 shows us three plots generated for shift $D \rightarrow W$ to assist us in analyzing P-ADC. In Figure 4a, we visualize the process of training and testing on Deep CORAL and P-ADC. We can see our method outperformed Deep CORAL on the test accuracies. Figure 4b shows the average loss in the training and test stage. It can be seen that our method was more stable in the test stage. Comparing Figure 4b,c, we can conclude that the p-norm loss was not always decreasing during training as the CORAL loss, but nevertheless, the two losses were about the same after training for hundreds of iterations. Furthermore, our p-norm loss could converge finally, constraining the distance between the source and target domain and maintaining an equilibrium in the target domain even more effectively than the CORAL loss.



(**a**) Training and test accuracies (**b**) Average loss (**c**) Classification loss

Figure 4. Comparative analysis of Deep CORAL and P-ADC$_{(2-4)}$ in shift $D \rightarrow W$. (**a**) Training and test accuracies on Deep CORAL and P-ADC$_{(2-4)}$. P-ADC$_{(2-4)}$ significantly outperforms Deep CORAL on the target domain under the same environmental setting. (**b**) Average loss value of Deep CORAL and P-ADC$_{(2-4)}$ in the training and test stage. (**c**) Classification loss value, p-norm loss value, and total loss value of P-ADC$_{(2-4)}$, respectively.

5. Conclusions

In this paper, we extended Deep CORAL, a simple, yet effective end-to-end adaptation in deep neural networks, with an attention mechanism to provide more information for deep neural networks. Meanwhile, we used the p-norm loss function to replace CORAL loss to balance the offset. Experiments on standard benchmark datasets (Office-31) showed state-of-the-art performance.

We tested our method on the classic benchmark dataset Office-31, and the experimental results showed us its effectiveness. One of the future research directions is the application of our method to a more diverse range of real-world applications and datasets. In addition, we are performing research on image recognition of vegetable diseases and insect pests under a greenhouse environment, which is very complicated. Different diseases and insect pests overlap, and light changes in real time. We hope to improve the accuracy

of vegetable disease and insect pest identification with a domain adaptation method, including this method.

Author Contributions: Conceptualization, Z.-Y.W.; methodology, Z.-Y.W.; software, Z.-Y.W.; validation, Z.-Y.W.; formal analysis, Z.-Y.W.; investigation, Z.-Y.W.; resources, D.-K.K.; data curation, Z.-Y.W.; writing—original draft preparation, Z.-Y.W.; writing—review and editing, D.-K.K.; visualization, Z.-Y.W.; supervision, D.-K.K.; project administration, D.-K.K.; funding acquisition, D.-K.K. All authors read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1D1A1A02050166).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [https://paperswithcode.com/dataset/office-31].

Acknowledgments: The authors wish to thank members of the Dongseo University Machine Learning/Deep Learning Research Laboratory and the anonymous referees for their helpful comments on earlier drafts of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015; pp. 1026–1034. doi:10.1109/ICCV.2015.123.
2. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature* **2016**, *529*, 484–489. doi:10.1038/nature16961.
3. Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature* **2020**, *588*, 604–609.
4. Hoffman, J.; Rodner, E.; Donahue, J.; Saenko, K.; Darrell, T. Efficient Learning of Domain Invariant Image Representations. In Proceedings of the 1st International Conference on Learning Representations (ICLR), Scottsdale, Arizona, 2–4 May 2013.
5. Luo, Y.; Zheng, L.; Guan, T.; Yu, J.; Yang, Y. Taking a Closer Look at Domain Shift: Category-Level Adversaries for Semantics Consistent Domain Adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
6. Zhao, S.; Yue, X.; Zhang, S.; Li, B.; Zhao, H.; Wu, B.; Krishna, R.; Gonzalez, J.E.; Sangiovanni-Vincentelli, A.L.; Seshia, S.A.; et al. A Review of Single-Source Deep Unsupervised Visual Domain Adaptation. *IEEE Trans. Neural. Netw. Learn. Syst.* **2020**. doi:10.1109/TNNLS.2020.3028503.
7. Sun, B.; Feng, J.; Saenko, K. Return of frustratingly easy domain adaptation. *arXiv* **2015**, arXiv:1511.05547.
8. Sun, B.; Saenko, K. Deep Coral: Correlation Alignment for Deep Domain Adaptation. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 443–450.
9. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105.
10. Wikipedia Contributors. Norm (Mathematics)— Wikipedia, The Free Encyclopedia. 2021. Available online: https://en.wikipedia.ahut.cf/w/index.php?title=Norm_(mathematics)&oldid=1020979856 (accessed on 19 May 2021).
11. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-attention generative adversarial networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 7354–7363.
12. Ramachandran, P.; Parmar, N.; Vaswani, A.; Bello, I.; Levskaya, A.; Shlens, J. Stand-alone self-attention in vision models. *arXiv* **2019**, arXiv:1906.05909.
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
14. Saenko, K.; Kulis, B.; Fritz, M.; Darrell, T. Adapting visual category models to new domains. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 213–226.
15. Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep domain confusion: Maximizing for domain invariance. *arXiv* **2014**, arXiv:1412.3474.

16. Long, M.; Cao, Z.; Wang, J.; Jordan, M.I. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*; NeurIPS: Montreal, QC, Canada, 3–8 December 2018; pp. 1640–1650.
17. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.F. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, USA, 20–25 June 2009; pp. 248–255.