Deep adversarial domain adaptation network

Lan Wu¹, Chongyang Li¹, Qiliang Chen²,³ and Binquan Li¹

Abstract
The advantage of adversarial domain adaptation is that it uses the idea of adversarial adaptation to confuse the feature distribution of two domains and solve the problem of domain transfer in transfer learning. However, although the discriminator completely confuses the two domains, adversarial domain adaptation still cannot guarantee the consistent feature distribution of the two domains, which may further deteriorate the recognition accuracy. Therefore, in this article, we propose a deep adversarial domain adaptation network, which optimises the feature distribution of the two confused domains by adding multi-kernel maximum mean discrepancy to the feature layer and designing a new loss function to ensure good recognition accuracy. In the last part, some simulation results based on the Office-31 and Underwater data sets show that the deep adversarial domain adaptation network can optimise the feature distribution and promote positive transfer, thus improving the classification accuracy.

Keywords
Domain adaptation, adversarial learning, adversarial domain adaptation, transfer learning, distribution difference, maximum average difference

Date received: 5 July 2020; accepted: 17 September 2020

Introduction
The training of deep neural networks requires a large amount of labelled data. However, the lack of a large amount of labelled data in real life limits its application. The proposal of transfer learning¹ solves the above problems well. It can apply knowledge or methods learned in other areas with large amounts of tagged data to areas where tagged data are scarce. Domain adaptation² is considered a representative method in transfer learning, which can use a considerable amount of labelled data from the related source domains to transfer the learned knowledge to the target domain.

The core idea of the domain adaptation method is to mitigate the problem of domain transfer between the source and the target domains. The domain-invariant features learned from the source domain with rich tagged data are transferred to the target domain. In recent years, in order to solve the problem of domain transfer between the source domain and the target domain, many domain adaptation methods have been proposed. The standard domain adaptation methods can be divided into the following two categories: (1) instance-based domain adaptation and (2) feature-representation domain adaptation.³ The instance-based method is used to slow the error by weighting the source samples and train the weighted source samples.⁴ The feature-based methods usually transform the features of the source and the target domains into a shared space

¹ School of Electrical Engineering, Henan University of Technology, Zhengzhou, China
² College of System Engineering, National University of Defence Technology, Changsha, China
³ Beijing Microelectronics Technology Institute, Beijing, China

Corresponding author:
Lan Wu, School of Electrical Engineering, Henan University of Technology, Zhengzhou 450001, China.
Email: wulan@haut.edu.cn
where the feature distributions of the two data sets match. The domain adaptive method, based on feature representation, is the most commonly used method.

The transfer component analysis\(^5\) (TCA) method was proposed previously, which utilises the feature extraction method and adopts the new parameter kernel for domain adaptation, projecting the data to the learned transfer component, which significantly reduces the distance between the domain distributions. In another study, a new kernel-based approach was proposed, which is a manifold learning method geodesic flow kernel (GFK), interpolating the bridge domain through infinite intermediate subspaces.\(^6\) A deep domain confusion\(^7\) (DDC) network structure was proposed, which was the first method to regularise the adaptive layer of AlexNet by using the linear kernel maximum mean discrepancy (MMD), to maximise the domain-invariant features and reduce the domain differences to realise domain adaptation. An extremely simple domain adaptive neural network\(^8\) (DaNN) was proposed; this is a new method for domain adaptation in deep architecture and involves the addition of an MMD adaptation layer after the feature layer to measure the difference in the feature distribution between two domains. A new deep adaptive network\(^9\) structure (DAN) was proposed, which extends deep convolutional neural networks to domain adaptation scenarios, copies the core Hilbert space reproducing kernel hilbert space (RKHS) by embedding the depth features of multiple task-specific layers and uses multi-kernel maximum mean discrepancy (MK-MMD) to optimise and match different distributions to learn transferable features. A new convolutional neural networks (CNN) structure was proposed, which simultaneously optimises domain invariants to promote domain migration and utilises soft tags to optimise the differences between the two tasks.\(^10\) A new domain adversarial neural network\(^11\) (DAN), was proposed, which was the first to introduce the idea of adversarial learning into transfer learning. The model minimises the loss function of the label classifier to extract features with distinguishing capabilities and maximises the loss function of a classifier to extract features with domain invariance, thereby reducing the domain migration. A deep network residual transfer network\(^12\) (RTN) domain adaptation method was proposed, which can learn adaptive classifiers and transfer features from the data in two domains. A weighted MMD model\(^13\) was proposed, which is defined by introducing an auxiliary weight for each class in the source domain, and a classification expectation-maximization (EM) algorithm based on pseudo-label assignment, auxiliary weight estimation and other updating of model parameters were proposed. The adversarial discriminative domain adaptation\(^14\) (ADDA) model was proposed, which combines generative adversarial networks (GAN)\(^15\) loss, weight sharing and discrimination modelling to optimise the domain differences. The joint adaptation network\(^16\) (JAN) was also proposed, which is based on the joint maximum mean discrepancy (JMMD) standard. The method learns the transfer network by aligning the joint average of several domain-specific layers across multiple domains and maximises the JMMD by adopting the adversarial training strategy to make the distribution of the two domains easy to distinguish.

The domain transfer problem is solved by the different methods mentioned above. However, there are still some problems that remain to be solved. The MMD is used to optimise the domain transfer between two domains. Although the distribution of these two domains is generally the same, we cannot guarantee the complete alignment of each category’s feature distribution, resulting in low classification accuracy. Domain adaptation using adversarial\(^17\) ideas is still a challenge, and even if the discriminator completely confuses the two domains, the feature distribution is not guaranteed to be sufficiently similar.

To further alleviate the distribution difference between the two domains, in this article, we propose deep adversarial domain adaptation network (DADAN), a new optimisation method. The MMD metric layer was added to the feature layer that was confused by the discriminator when adversarial training was used. In addition, weights of the class were established for each category to participate in the training, to maximise the differences between classes and to minimise the differences within classes, thereby improving classification accuracy.

The rest of this article is organised as follows: the second section introduces the related work; the third section explains the model architecture; the fourth section shows the experimental content, the simulation results and the analysis of the results; and finally, the fifth section concludes this article.

**Related work**

**Maximum mean discrepancy**

The MMD is usually the square distance between the kernel embedding of data distributions \(P_s\) and \(P_t\) in the RKHS. The smaller the MMD value is, the more similar are the feature distributions of the two domains. The distribution difference between the two domains can be estimated in many ways. The most commonly used non-parametric method used in transfer learning to measure the distribution difference between two domains is MMD. It measures the distance between two distributions in RKHS by a characteristic kernel \(K\) that maps the original variable to the RKHS space.

The characteristic distribution distance between the source domain and the target domain can be expressed as follows

\[
\text{MMD}^2_k(P_s, P_t) = \left\| E_{P_s} [\varphi(x_s)] - E_{P_t} [\varphi(x_t)] \right\|^2_{H_k}
\]  (1)
Proposed method

The quality of the extracted deep domain-invariant features will influence the model’s performance and reduce the difference in the feature distribution between the two domains that are confused by the domain discriminator and improve the performance of the model. In this article, we propose the combination of PADA and MK-MMD to further reduce the distribution difference between the two domains, confused by the domain discriminator. When the distribution difference is smaller, the larger the amount of useful information carried by the extracted domain-invariant features is, the better is the transfer effect.

In this article, we propose to add the MK-MMD measurement layer on the feature layer extracted by the feature extractor and confused by the domain discriminator and add its loss to the overall optimisation goal of the model, thereby further reducing the difference in the feature distribution of the two domains confused by the domain discriminator.

Network structure

We propose a new DADAN model structure. After pre-processing the collected original image data, the result was input into the network for model training. The network structure is shown in Figure 1.

The feature extractor $G_f$ discussed in this article was obtained by fine-tuning the pre-trained ResNet-50 in the ImageNet model, which could make the most of the advantages of the pre-trained model and the original network parameters. The deep domain-invariant features $f$ of the input data were extracted by the feature extractor $G_f$, and the weight of the feature extractor was shared between the two domains. Here, $M_s$ represents the features extracted from the source domain, and $M_t$ denotes the features extracted from the target domain.

The source classifier $G_s$ took the domain-invariant features $f$ obtained from the feature extractor $G_f$ as the input for detection and shared the weight of the source classifier to target the domain classifier.

The domain discriminator $G_d$ was used to confuse the domain-invariant feature $f$ of the two domains extracted by the feature extractor $G_f$. The gradient reversal layer acted between the feature extractor $G_f$ and the domain discriminator $G_d$, and thus, the gradient direction was automatically reversed during the back-propagation process.

In order to obtain the domain-invariant feature $f$, we learnt the parameter $\theta_f$ of the feature extractor $G_f$ by maximising the loss of the domain discriminator $G_d$ and learnt the parameter $\theta_g$ of the domain discriminator $G_d$ by minimising the loss of the domain discriminator. In order to ensure the accuracy of the source classifier, the loss of the source classifier $G_s$ was also minimised. The parameter $\theta_m$ of MK-MMD was minimised to further reduce the

\[
\text{MMD}_k^2(P_s, P_t) = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \varphi(x_i^s) - \frac{1}{n_t} \sum_{i=1}^{n_t} \varphi(x_i^t) \right\|_{H_k}^2 
= \frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} k(x_i^s, x_j^s) - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} k(x_i^s, x_j^t) 
+ \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} k(x_i^t, x_j^t)
\]

where $\varphi$ is the feature mapping and $H_k$ stands for RKHS with the feature kernel $k$.

Partial adversarial domain adaptation

Partial adversarial domain adaptation (PADA)\textsuperscript{18} can be improved using the traditional DANN network. We are given a source domain $D_s = \{x_i^s, y_i^s\}_{i=1}^{n_s}$ of $n_s$ labelled examples associated with $C_s$ classes and a target domain $D_t = \{x_i^t\}_{i=1}^{n_t}$ of $n_t$ unlabelled examples associated with $C_t$ classes. $P(x_i^s, y_i^s)$ and $Q(x_i^t, y_i^t)$ represent the joint distribution of the source domain and the target domain, respectively, and $P \neq Q$. We define $C_s > C_t$ to represent the partial domain adaptation and $C_s = C_t$ to represent the traditional domain adaptation.

The PADA method can promote the positive transfer of the relevant source domain and slow down the negative transfer of the irrelevant source domain. The overall objective of PADA\textsuperscript{18} is as follows

\[
C_0(\theta_f, \theta_g, \theta_d) = \frac{1}{n_s} \sum_{x_i \in D_s} \gamma_i L_y(G_f(G_f(x_i)), y_i) 
- \frac{\lambda}{n_t} \sum_{x_i \in D_t} \gamma_i L_g(G_y(G_y(x_i)), d_i) 
- \frac{\lambda}{n_t} \sum_{x_i \in D_t} L_d(G_d(G_d(x_i)), d_i)
\]

where $y_i$ is the original label of the source domain sample and $d_i$ is the domain tag of input $x_i$, and $\gamma$ is used to represent the overall distribution weight of the source domain data. The $G_f$ represents the feature extractor, the $G_s$ represents the source classifier and the $G_d$ represents the domain discriminator.

The PADA method has achieved good results in partial domain adaptation.\textsuperscript{18, 20} However, in the case of the traditional domain adaptation, the data feature distribution of the domain discriminator confusion cannot be guaranteed to be the same; therefore, in the present study, we improved this. In this article, we propose to add an MMD measurement layer to the domain-invariant feature extracted by the feature extractor and add it to the optimisation target. Further optimisation of the characteristic distribution of the two domains is confused by the domain discriminator.
MK-MMD\textsuperscript{9,21} was developed from MMD. The original MMD maps the features of two domains to an RKHS and calculates the average difference after mapping. This kernel is fixed; thus, we can choose either a Gaussian kernel or a linear kernel, but we cannot determine which kernel is necessarily better. Therefore, in this study, we adopted MK-MMD. The MK-MMD model assumes that we can obtain the optimal kernel through a linear combination of multiple kernels. The most popular method of using MK-MMD is DAN.\textsuperscript{9}

\section*{Loss function}

1. Source supervision loss

The feature $f$ extracted from the feature extractor $G_f$ was fed into the source classifier $G_s$ to optimise the classification loss of the labelled data in the source domain. The loss function can be expressed as follows

$$\min_{M_s, G_s} L_{G_s}(X_s, Y_s) = -E_{(x_i, y_i) \sim (X_s, Y_s)} \sum_{k=1}^{K} I_{[y_k = y_i]} \log G_s(M_s(x_i))$$

(4)

The source classifier $\hat{y}_i = G_s(x_i)$ could well represent the probability distribution of each input data item $x_i$ in the source label space. When the target domain data in the same label space were used to predict the classification result by the source classifier, some of them were more likely to be assigned to other classes, resulting in classification errors. In order to reduce these classification errors, we set the contribution weights for each category to improve the training accuracy. This setting could also reduce the probability of allocating the target domain data to other classes. The source class weight contribution parameters were as follows

$$\gamma = \frac{1}{m_t} \sum_{i=1}^{m_t} \hat{y}_i$$

(5)

where $\gamma$ is a vector that measures the class contribution of the source domain. Irrespective of whether the number of categories in the source domain was equal to the number of target domains, the contribution weight of the source domain categories could be obtained, thus guiding the target domain data to reduce the classification errors.

2. Domain discriminator loss

The domain discriminator $G_d$ was used to identify which domain the domain-invariant features extracted from each input data item $x_i$ originated from. If the domain discriminator $G_d$ could not distinguish which domain the extracted domain-invariant features belonged to, then the features extracted by the feature extractor $G_f$ were domain invariant. The loss function can be written as follows

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{DADAN_network.png}
\caption{DADAN network structure. DADAN: deep adversarial domain adaptation network.}
\end{figure}
min \ Gd \ (X, X_t, M_s, M_t) = -E_{(x_s) \sim (X, \log Gd(M_s(x_s)))}
- E_{(x_t) \sim (X, \log (1 - Gd(M_t(x_t)))}

\min_{M_s, M_t} \ L_M (X, X_t, G_d) = -E_{(x_s) \sim (X, \log Gd(M_t(x_t)))}

Here, \ LGd is used to train the domain discriminator to identify which domain the input data originated from. \ LM is used to confuse the extracted features so that the discriminator cannot identify which domain the data originated from.

3. Maximum average difference loss

MK-MMD was the same as MMD; the feature map \( \varphi \) was associated with the characteristic kernel, and \( k(x_s, x_t) = \langle \varphi(x_s), \varphi(x_t) \rangle \), \( k(x_s, x_t) \) was defined as a convex combination of \( \{k_u\} \) of \( m \) kernels as follows

\[ K = \sum_{i=1}^{m} \beta_u k_u : \sum_{i=1}^{m} \beta_u = 1, \beta_u \geq 0, \forall u \]

The correlation constraint of the coefficient \( \{\beta_u\} \) was used to ensure that the generated multi-kernel \( k \) was unique

\[ \min_{\text{MMD}_{k}} \ L_{\text{MMD}} = \text{MMD}_{k}(M_s, M_t)^2 \]

The overall objectives of DADAN were as follows

\[ C_0 (\theta_f, \theta_y, \theta_d, \theta_m) = \frac{1}{n_s} \sum_{x_s \in D_s} \gamma_y L_{Gd}(G_y(G_f(x_s)), y_i) \]

\[ - \frac{\lambda}{n_t} \sum_{x_t \in D_t} \gamma_y L_{Gd}(G_d(G_f(x_t)), d_i) \]

\[ - \frac{\lambda}{n_t} \sum_{x_t \in D_t} L_{\text{MMD}}(G_f(x_s), G_f(x_t)) \]

Here, \( y_i \) is the original label of the source domain sample and \( d_i \) is the domain from which the sample originated. Furthermore, \( \lambda \) is a hyper-parameter that trades off \( L_{Gd} \) and \( L_{G_d} \).
The optimisation problem found the optimal parameters $\theta_f$, $\theta_y$, $\theta_d$ and $\theta_m$ as follows:

$$(\hat{\theta}_f, \hat{\theta}_y) = \arg\min_{\theta_f, \theta_y} C_0(\theta_f, \theta_y, \theta_d, \theta_m)$$

$$(\hat{\theta}_d) = \arg\max_{\theta_d} C_0(\theta_f, \theta_y, \theta_d, \theta_m)$$

$$(\hat{\theta}_m) = \arg\min_{\theta_m} C_0(\theta_f, \theta_y, \theta_d, \theta_m)$$

\[ (11) \]

**Simulation**

We conducted an experimental validation to evaluate our approach and several advanced deep transfer learning methods. Our method was simulated with an underwater object detection task.

**Data set description**

1. Office-31 data set

*Office-31* is one of the most commonly used image data sets in domain adaptation, containing a total of 4652 images in 31 categories from three different domains: Amazon (A), Webcam (W) and DSLR (D). We evaluated the accuracy of the following six transfer tasks in different
methods: A → W, D → W, W → D, A → D, D → A and W → A.

2. Underwater data set

According to the method of generating the underwater data set proposed in Yu et al.,[23] three types of Office-31 underwater data sets with a turbidity of 0.5, 1.0 and 2.0 were obtained, as shown in Figure 2. In this study, the Office-31 data set and the generated underwater data set were taken as the source and the target domains. We evaluated nine transfer tasks for each turbidity. They were A → U_A, A → U_D, A → U_W, D → U_A, D → U_D, D → U_W, W → U_A, W → U_D and W → U_W.

Compared approaches and results analysis

For the Office-31 data set, we compared the traditional and advanced transfer learning and deep learning methods: TCA,[5] GFK,[6] DDC,[7] DAN,[9] RTN,[12] DANN,[11] ADDA[14] and JAN.[16] AlexNet and ResNet are both classic networks and are widely used in image classification. We adopted AlexNet[24] and ResNet-50[25] as the base networks. On the one hand, for the convenience of comparison with other mainstream methods; on the other hand, these two network structures can obtain better results in image feature extraction.

As the amount of data in domain A was considerably larger than that in domains W and D, it was impossible to learn more features in the W and D domains for tasks W → A and D → A, resulting in a less accurate result than that for other tasks.

Tables 1 and 2 show the AlexNet-based and ResNet-based results of different methods in the Office-31 data set, respectively. These tables show that our approach had the best results for different transfer tasks. Furthermore, for all of the transfer learning tasks, the proposed method was obviously superior to the other methods. The boldface values in the tables are the best results.

Figure 4. Average accuracy (%) of the Office-31 data set based on ResNet.

Figure 5. Accuracy (%) on the Underwater data set for the DADAN method. DADAN: deep adversarial domain adaptation network.
Figures 3 and 4 show the average accuracy of different transfer learning methods based on AlexNet and ResNet. It was evident that the DADAN method proposed in this article was superior to all of the comparison methods for all of the transfer learning tasks.

For the underwater image data set, the DADAN method proposed in this article was compared with the AlexNet-based method.23 As the amount of data in domain A was considerably larger than that in domains U_W and U_D, it was impossible to learn more features in the U_W and U_D domains for tasks \( W \rightarrow U_A \) and \( D \rightarrow U_A \), resulting in a less accurate result than that for the other tasks.

The AlexNet-based method23 only verified the accuracy of task \( A \rightarrow U_A \) under three different turbidity values and the maximum accuracy of less than 50\%. Figure 5 clearly shows that the DADAN method was considerably more accurate than the AlexNet-based method in the case of the transfer task \( A \rightarrow U_A \) under three different types of turbidity. In addition to the migration task \( A \rightarrow U_A \), Figure 5 shows the results of DADAN facing different turbidity values for different transfer learning tasks. We observed that the DADAN method could achieve not only high accuracy for the same transfer task under different turbidity conditions but also stable accuracy for different transfer tasks under different turbidity conditions. Compared with the previous methods, the DADAN method dramatically improved the accuracy of the transfer of in-air image data knowledge for underwater target detection tasks. Moreover, it had strong applicability and provided more possibilities for underwater target detection.

Conclusion

In this article, a new DADAN was proposed. On the basis of the features and advantages of adversarial learning and MK-MMD, DADAN put them together and designed a new loss function to further optimise the feature distribution of two domains in order to maintain their consistency. We conducted some comparative experiments on the Office-31 and Underwater data sets by using DAN, RTN, DANN and other methods. The experimental results showed that this method could effectively optimise the feature distribution confused by the domain discriminator, thus promoting the positive transfer and obtaining a higher accuracy classification than the current mainstream methods.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This study was supported by the National Natural Science Foundation of China, No. 61973103, Henan province Central plains thousand talents plan: top young talents and Key scientific research project of Henan University with No. 19A120002.

ORCID iDs

Lan Wu https://orcid.org/0000-0002-2497-6556
Chongyang Li https://orcid.org/0000-0002-6514-1774
Binquan Li https://orcid.org/0000-0003-3466-2523

References

1. Pan SJ and Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng 2010; 22: 1345–1359.
2. Wang M and Deng WH. Deep visual domain adaptation: a survey. Neurocomputing 2018; 312: 135–153.
3. Bai J, Cao R, Ma W, et al. Combination of feature-based and instance-based methods for domain adaptation in sentiment classification. In: 2019 international conference on technologies and applications of artificial intelligence (TAAI), Kaohsiung, Taiwan, 21–23 November 2019.
4. Di Gangi MA, Nguyen V-N, Negri M, et al. Instance-based Model Adaptation for Direct Speech Translation. In: 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2020, 4–8 May 2020, Barcelona, Spain, pp. 7914–7918. Institute of Electrical and Electronics Engineers Inc.
5. Pan SJ, Tsang IW, Kwok JT, et al. Domain adaptation via transfer component analysis. In: 21st international joint conference on artificial intelligence, IJCAI 2009, 11 July 2009–16 July 2009, Pasadena, CA, pp. 1187–1192.
6. Gong B, Shi Y, Sha F, et al. Geodesic flow kernel for unsupervised domain adaptation. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012, 16 June 2012–21 June 2012, Providence, RI, United States, pp. 2066–2073. IEEE Computer Society.
7. Tzeng E, Hoffman J, Zhang N, et al. Deep domain confusion: maximizing for domain invariance. arXiv e-prints, https://aiadsabs.harvard.edu/abs/2014arXiv1412.3474T (2014, accessed 1 December 2014).
8. Ghifary M, Bastiaan Kleijn W and Zhang M. Domain adaptive neural networks for object recognition. In: Pacific Rim international conference on artificial intelligence, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014, vol. 8862, pp. 898–904. DOI: 10.1007/978-3-319-13560-1.
9. Long M, Cao Y, Wang J, et al. Learning transferable features with deep adaptation networks. In: 32nd international conference on machine learning, ICML 2015, 6 July 2015–11 July 2015, Lille, France, 2015, pp. 97–105. International Machine Learning Society (IMLS).
10. Tzeng E, Hoffman J, Darrell T, et al. Simultaneous deep transfer across domains and tasks. In: 15th IEEE International Conference on Computer Vision, ICCV 2015, 11–18 December 2015, Santiago, Chile, pp. 4068–4076. Institute of Electrical and Electronics Engineers Inc.
11. Ganin Y, Ustinova E, Ajakan H, et al. Domain-adversarial training of neural networks. *J Mach Learn Res* 2016; 17: 1–35.

12. Long M, Zhu H, Wang J, et al. Unsupervised domain adaptation with residual transfer networks. In: *30th annual conference on neural information processing systems, NIPS 2016*, 5 December 2016–10 December 2016, Barcelona, Spain, pp. 136–144. Neural Information Processing Systems Foundation.

13. Yan H, Ding Y, Li P, et al. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In: *30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 21–26 July 2017, Honolulu, HI, United states, pp. 945–954. Institute of Electrical and Electronics Engineers Inc.

14. Tzeng E, Hoffman J, Saenko K, et al. Adversarial discriminative domain adaptation. In: *30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 21–26 July 2017, Honolulu, HI, United states, pp. 2962–2971. Institute of Electrical and Electronics Engineers Inc.

15. Goodfellow IJ, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks. *Adv Neural Inf Process Syst* 2014; 3: 2672–2680.

16. Long M, Zhu H, Wang J, et al. Deep transfer learning with joint adaptation networks. In: *34th International Conference on Machine Learning, ICML 2017*, 6–11 August 2017, Sydney, NSW, Australia, pp. 3470–3479. International Machine Learning Society (IMLS).

17. Shen T, Gong D, Zhang W, et al. Regularizing proxies with multi-adversarial training for unsupervised domain-adaptive semantic segmentation. *Comp Sci* 2019; abs/1907.12282.

18. Cao Z, Ma L, Long M, et al. Partial adversarial domain adaptation. In: *15th European Conference on Computer Vision, ECCV 2018*, 8–14 September 2018, Munich, Germany, pp. 139–155. Springer Verlag.

19. Cao Z, Long M, Wang J, et al. Partial Transfer Learning with Selective Adversarial Networks. In: *31st Meeting of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018*, 18–22 June 2018, Salt Lake City, UT, United states, pp. 2724–2732. IEEE Computer Society.

20. Zhang J, Ding Z, Li W, et al. Importance Weighted Adversarial Nets for Partial Domain Adaptation. In: *31st Meeting of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 18–22 June 2018, Salt Lake City, UT, United states, pp. 8156–8164. IEEE Computer Society.

21. Ling W and Han S. Deep Adaptation Network Combining Domain Confusion With MK-MMD. *J Chin Comput Syst* 2019; 1519–1524.

22. Saenko K, Kulis B, Fritz M, et al. Adapting visual category models to new domains. In: *11th European Conference on Computer Vision, ECCV*, 10–11 September 2010, Heraklion, Crete, Greece, pp. 213–226. Springer Verlag.

23. Krizhevsky A, Sutskever I and Hinton GE. ImageNet classification with deep convolutional neural networks. In: *26th Annual Conference on Neural Information Processing Systems 2012, NIPS 2012*, 3–6 December 2012, Lake Tahoe, NV, United states, pp. 1097–1105. Neural information processing systems foundation.

24. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: *29th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, 26 June – 1 July 2016, Las Vegas, NV, United states, pp. 770–778. IEEE Computer Society.

25. Yu X, Xing X, Zheng H, et al. Man-Made Object Recognition from Underwater Optical Images Using Deep Learning and Transfer Learning. In: *2018 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2018*, 15–20 April 2018, Calgary, Alberta, Canada, pp. 1852–1856. Institute of Electrical and Electronics Engineers Inc.