

Adversarial Consistent Learning on Partial Domain Adaptation of PlantCLEF 2020 Challenge

Youshan Zhang and Brian D. Davison

Lehigh University, Computer Science and Engineering, Bethlehem, PA, USA
{yoz217,bdd3}@lehigh.edu

Abstract. Domain adaptation is one of the most crucial techniques to mitigate the domain shift problem, which exists when transferring knowledge from an abundant labeled sourced domain to a target domain with few or no labels. Partial domain adaptation addresses the scenario when target categories are only a subset of source categories. In this paper, to enable the efficient representation of cross-domain plant images, we first extract deep features from pre-trained models and then develop adversarial consistent learning (*ACL*) in a unified deep architecture for partial domain adaptation. It consists of source domain classification loss, adversarial learning loss, and feature consistency loss. Adversarial learning loss can maintain domain-invariant features between the source and target domains. Moreover, feature consistency loss can preserve the fine-grained feature transition between two domains. We also find the shared categories of two domains via down-weighting the irrelevant categories in the source domain. Experimental results demonstrate that training features from NASNetLarge model with proposed *ACL* architecture yields promising results on the PlantCLEF 2020 Challenge.

Keywords: Adversarial learning · Partial domain adaptation · Plant identification.

1 Introduction

Automated plant identification is important in recognizing plant species. The availability of massive labeled training data is a prerequisite of machine learning models. Unfortunately, such a requirement cannot be met in the plant identification problem since we have sparse labels for real-world plant images. Therefore, we propose to transfer knowledge from an existing auxiliary labeled herbarium domain to the field photo domain with limited or no labels. However, due to the phenomenon of data bias or domain shift [11], classification models do not generalize well from an existing herbarium domain to a novel field photo domain.

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). CLEF 2020, 22-25 September 2020, Thessaloniki, Greece.

Domain adaptation (DA) has been proposed to leverage knowledge from an abundant labeled source domain to learn an effective predictor for the target domain with few or no labels, while mitigating the domain shift problem [16,17,19,20]. In this paper, we focus on unsupervised domain adaptation (UDA), where the target domain has no labels. Since we have fewer classes in the field photo domain, and the classes of the field photo domain is a subset of the classes of the source herbarium domain, we investigate partial domain adaptation (PDA) for the PlantCLEF 2020 Challenge.

Recently, deep neural network methods have been widely used in the domain adaptation problem. Notably, adversarial learning shows its power in embedding in deep neural networks to learn feature representations to minimize the discrepancy between the source and target domains [9,14]. Inspired by the generative adversarial network (GAN) [6], adversarial learning also contains a feature extractor and a domain discriminator. The domain discriminator can distinguish the source domain from the target domain, while the feature extractor can learn domain-invariant representations to fool the domain discriminator [9,10,18]. The target domain risk (the error of the target domain) is expected to be minimized via minimax optimization. Cao et al. presented adversarial learning for PDA, which alleviates negative transfer by reducing the outlier of source classes for training the source classifier and domain labels, while positive transfer is improved via matching the feature distributions in the shared label space [2]. Similarly, the example transfer network is proposed to jointly learn domain-invariant representations and a progressive weighting method to examine the transferability of source examples. The model can improve positive transfer by relevant examples and mitigate negative transfer by identifying irrelevant examples [3].

Although many methods are proposed for partial domain adaptation, they still suffer from two challenges: (1) the models are evaluated on small datasets, while it has lower transferability to the large-scale dataset, and (2) the feature consistency of two domains is inappropriately ignored.

To address the aforementioned challenges, we aggregate three different loss functions in one framework: source domain classification loss, adversarial learning loss, and feature consistency loss to reduce the discrepancy of the two domains. Moreover, our model is evaluated on a large-scale plant identification dataset to improve the estimate of the generalization ability of our model.

Our contributions are three-fold:

1. We propose a novel adversarial consistent learning network (*ACL*) for PDA, to adversarially minimize the domain discrepancy of the source and target domains and maintain domain-invariant features;
2. The proposed adversarial learning loss and feature consistency loss can distinguish the target domain from the source domain, and preserve the fine-grained feature transition between the two domains;
3. We impose shared category selection to filter out the irrelevant categories in the source domain. By down-weighting the irrelevant categories in the source domain, we can reduce negative transfer from the source domain to the target domain.



Fig. 1: Examples images of the PlantCLEF 2020 dataset. The large discrepancy between training and test data cause the difficulty in the PDA.

Experimental results show that *ACL* achieves higher classification accuracy than several baseline methods and yields promising results on the PlantCLEF 2020 Challenge.

2 Dataset

PlantCLEF 2020 is a large-scale dataset of the PlantCLEF 2020 task [5], organized in the context of the LifeCLEF 2020 challenge [8]. Fig. 1 shows some challenging images in this dataset. The herbarium domain contains 320,750 images in 997 species, and the number of images in different species are unbalanced. This dataset consists of herbarium sheets whereas the test set will be composed of field pictures. The validation set consists of two domains herbarium_photo_associations and photos. Herbarium_photo_associations domain includes 1,816 images from 244 species. This domain contains both herbarium sheets and field pictures for a subset of species, which enables learning a mapping between the herbarium sheets domain and the field pictures domain. Another photo domain has 4,482 images from 375 species and images are from plant pictures in the field, which is similar to the test dataset. The test dataset contains 3,186 unlabeled images. Due to the significant difference between herbarium and real photos, it is extremely difficult to identify the correct class.

We exclude the classId of “108335” in the photo domain since the major classes are from the herbarium domain. In addition, herbarium domain does not contain the “108335” category. Therefore, eight images are excluded in the photo domain. The statistics of the PlantCLEF 2020 dataset are listed in Tab. 1.

Table 1: Statistics on PlantCLEF 2020 dataset

	Domain	Number of Samples	Number of Classes
	Herbarium (H)	320,750	997
Herbarium_photo_associations (A)		1,816	244
	Photo (P)	4,482	375
	Test (T)	3,186	-

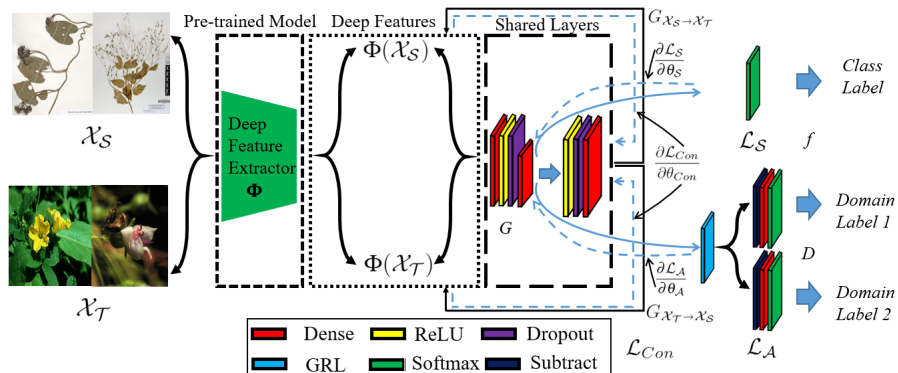


Fig. 2: The architecture of our proposed *ACL* model. We first extract deep features from a pre-trained model for both source and target domains via Φ . The shared layers are jointly trained with source and target features. Also, the parameters in shared layers are updated by the backward gradients ($\frac{\partial \mathcal{L}_S}{\partial \theta_S}$, $\frac{\partial \mathcal{L}_A}{\partial \theta_A}$ and $\frac{\partial \mathcal{L}_{Con}}{\partial \theta_{Con}}$) from class label classifier, domain label predictor and feature consistency regressor. The *ACL* model consists of three different loss functions (source classification loss \mathcal{L}_S , adversarial domain loss \mathcal{L}_A , and feature consistency loss \mathcal{L}_{Con}). The feature extractor G in the shared layers is used for both classifier f and domain discriminator D (The blue dash lines are the backward gradients, and GRL stands for gradient reversal layer). Layers visualization of architecture is shown in Fig. 3.

3 Methods

3.1 Motivation

Previous partial domain adaptation methods [2, 3] evaluated their models based on a small dataset (e.g., Office 31), while their models have lower generalizability to large-scale datasets. In addition, feature consistency of both source and target domains is not well addressed in the PDA.

In this paper, we present our approach: adversarial consistent learning (ACL) on partial domain adaptation. It can align the feature distribution of the source and target domains in the shared categories and guarantee feature consistency across the two domains. Importantly, ACL identifies irrelevant source categories via down-weighting class importance automatically. Evaluation on the large-scale PlantCLEF 2020 challenge dataset shows a high generalizability of our model.

3.2 Problem and notation

For unsupervised domain adaptation, given a source domain $\mathcal{D}_S = \{\mathcal{X}_{S_i}, \mathcal{Y}_{S_i}\}_{i=1}^{\mathcal{N}_S}$ of \mathcal{N}_S labeled samples across the set of categories \mathcal{C}_S and a target domain $\mathcal{D}_T = \{\mathcal{X}_{T_j}\}_{j=1}^{\mathcal{N}_T}$ of \mathcal{N}_T samples without any labels (\mathcal{Y}_T is unknown) across the set of

categories $\mathcal{C}_{\mathcal{T}}$. For partial domain adaptation, the number of categories in $\mathcal{C}_{\mathcal{T}}$ is less than the number of categories in $\mathcal{C}_{\mathcal{S}}$, and $\mathcal{C}_{\mathcal{T}} \subsetneq \mathcal{C}_{\mathcal{S}}$. The samples $\mathcal{X}_{\mathcal{S}}$ and $\mathcal{X}_{\mathcal{T}}$ obey the marginal distributions of $P_{\mathcal{S}}$ and $P_{\mathcal{T}}$. The conditional distributions of two domains are denoted as $Q_{\mathcal{S}}$ and $Q_{\mathcal{T}}$. Due to the discrepancy of two domains, the distributions are assumed to be different, i.e., $P_{\mathcal{S}} \neq P_{\mathcal{T}}$ and $Q_{\mathcal{S}} \neq Q_{\mathcal{T}}$. Our ultimate goal is to learn a classifier f under a feature extractor G , which selects shared categories between two domains, and ensures lower generalization error in the target domain.

3.3 Deep features extraction

To circumvent the large computation resource requirement of training large-scale PlantCLEF 2020 challenge datasets, we instead focus on deep features from pre-trained models. Based on Zhang and Davison [17], the deep features are extracted from the last fully connected layer of the pretrained model via Φ . One represented feature vector has the size of 1×1000 and corresponds to one plant image. Therefore, the source domain and the target domain can be represented by $\Phi(\mathcal{X}_{\mathcal{S}}) \in \mathbb{R}^{\mathcal{N}_{\mathcal{S}} \times 1000}$ and $\Phi(\mathcal{X}_{\mathcal{T}}) \in \mathbb{R}^{\mathcal{N}_{\mathcal{T}} \times 1000}$, respectively.

3.4 Source classifier

The task in the source domain is trained using the typical cross-entropy loss in following equation:

$$\mathcal{L}_{\mathcal{S}}(f(G(\Phi(\mathcal{X}_{\mathcal{S}}))), \mathcal{Y}_{\mathcal{S}}) = -\frac{1}{\mathcal{N}_{\mathcal{S}}} \sum_{i=1}^{\mathcal{N}_{\mathcal{S}}} \sum_{c=1}^{\mathcal{C}_{\mathcal{S}}} \mathcal{Y}_{\mathcal{S}_{ic}} \log(f(G(\Phi(\mathcal{X}_{\mathcal{S}_i}))), \quad (1)$$

where $\mathcal{Y}_{\mathcal{S}_{ic}} \in [0, 1]^{\mathcal{C}_{\mathcal{S}}}$ is the probability of each class of ground truth for the i th element of \mathcal{S} , f is the classifier in Fig. 2, and $f(G(\Phi(\mathcal{X}_{\mathcal{S}_i})))$ is the predicted probability.

3.5 Adversarial domain loss

In general adversarial learning, the system learns a mapping from the source domain to the target domain. Given the feature representation of feature extractor G , we can learn a discriminator D , which can distinguish the two domains using the following loss function:

$$\begin{aligned} \mathcal{L}_{\mathcal{A}}(G_{\mathcal{X}_{\mathcal{S}} \rightarrow \mathcal{X}_{\mathcal{T}}}, G(\Phi(\mathcal{X}_{\mathcal{S}})), G(\Phi(\mathcal{X}_{\mathcal{T}}))) &= -\frac{1}{\mathcal{N}_{\mathcal{S}}} \sum_{i=1}^{\mathcal{N}_{\mathcal{S}}} \log(1 - D(G(\Phi(\mathcal{X}_{\mathcal{S}_i}))) \\ &\quad - \frac{1}{\mathcal{N}_{\mathcal{T}}} \sum_{j=1}^{\mathcal{N}_{\mathcal{T}}} \log(D(G(\Phi(\mathcal{X}_{\mathcal{T}_j}))). \end{aligned} \quad (2)$$

However, Eq. 2 only guarantees source domain data will be close to the target data ($G_{\mathcal{X}_{\mathcal{S}} \rightarrow \mathcal{X}_{\mathcal{T}}}$), and it does not ensure that the target data will be close to the

source data. We hence introduce another mapping from the target domain to the source domain $G_{\mathcal{X}_T \rightarrow \mathcal{X}_S}$ in Eq. 3 and train it with the same adversarial loss as in $G_{\mathcal{X}_S \rightarrow \mathcal{X}_T}$ as shown in Eq. 2.

$$\mathcal{L}_{\mathcal{A}}(G_{\mathcal{X}_T \rightarrow \mathcal{X}_S}, G(\Phi(\mathcal{X}_S)), G(\Phi(\mathcal{X}_T))) \quad (3)$$

For $G_{\mathcal{X}_S \rightarrow \mathcal{X}_T}$, the source domain has the label of 0 and the target domain has the label of 1, which is corresponding to *Domain Label 1* in Fig. 2. Meanwhile, for $G_{\mathcal{X}_T \rightarrow \mathcal{X}_S}$, 1 is the new label for the source domain and 0 is the new label for target domain, which is corresponding to *Domain Label 2* in Fig. 2. Therefore, we define the adversarial learning loss as:

$$\begin{aligned} \mathcal{L}_{\mathcal{A}}(G(\Phi(\mathcal{X}_S)), G(\Phi(\mathcal{X}_T))) &= \mathcal{L}_{\mathcal{A}}(G_{\mathcal{X}_S \rightarrow \mathcal{X}_T}, G(\Phi(\mathcal{X}_S)), G(\Phi(\mathcal{X}_T))) \\ &+ \mathcal{L}_{\mathcal{A}}(G_{\mathcal{X}_T \rightarrow \mathcal{X}_S}, G(\Phi(\mathcal{X}_S)), G(\Phi(\mathcal{X}_T))). \end{aligned} \quad (4)$$

3.6 Feature consistency loss

To encourage the source domain and target domain information to be preserved during adversarial learning, we propose a feature consistency loss in our model. Details of the feature reconstruction layers are shown in Fig. 2; the reconstructed layers are right behind the feature extractor G in the shared layers, and they aim to reconstruct the extracted features and maintain the invariant features during the conversion process. The feature consistency loss is defined as:

$$\begin{aligned} \mathcal{L}_{Con}(G_{\mathcal{X}_S \rightarrow \mathcal{X}_T}, G_{\mathcal{X}_T \rightarrow \mathcal{X}_S}, G(\Phi(\mathcal{X}_S)), G(\Phi(\mathcal{X}_T))) \\ = \mathbb{E}_{x_s \sim G(\Phi(\mathcal{X}_S))} [\ell(G_{\mathcal{X}_T \rightarrow \mathcal{X}_S}(G_{\mathcal{X}_S \rightarrow \mathcal{X}_T}(x_s)) - x_s)] \\ + \mathbb{E}_{x_t \sim G(\Phi(\mathcal{X}_T))} [\ell(G_{\mathcal{X}_S \rightarrow \mathcal{X}_T}(G_{\mathcal{X}_T \rightarrow \mathcal{X}_S}(x_t)) - x_t)], \end{aligned} \quad (5)$$

where ℓ is the mean squared error loss function, which calculates the difference between true features and the reconstructed features.

3.7 Shared categories selection

In PDA, the set of target domain labels is a subset of the source domain labels, i.e., $\mathcal{C}_T \subsetneq \mathcal{C}_S$. In the PlantCLEF challenge, the size of irrelevant label set ($\mathcal{C}_S - \mathcal{C}_T$) is far larger than the size of \mathcal{C}_T ($|\mathcal{C}_S - \mathcal{C}_T| \gg |\mathcal{C}_T|$). If we use all elements of the source domain distribution to match the target domain distribution, it will cause negative transfer since the target domain will also be forced to match the irrelevant labels ($\mathcal{C}_S - \mathcal{C}_T$). Therefore, it is important to identify the shared categories between source and target domains.

To address the aforementioned challenge, we re-weight the source domain label set via reducing the irrelevant label set. During the training, we can get the predicted probability of the target domain: $\mathcal{Y}_{T_j} = f(G(\Phi(\mathcal{X}_{T_j})))$, which gives a probability of each source label in \mathcal{C}_S . As we know, the set of irrelevant source labels and target label set are disjoint, and the target data are significantly dissimilar to the source data in the irrelevant label set. Therefore, the probability

of irrelevant categories should be sufficiently small and can be ignored. We then defined the weight vector as:

$$\mathcal{W} = \frac{1}{\mathcal{N}_{\mathcal{T}}} \sum_{j=1}^{\mathcal{N}_{\mathcal{T}}} \mathcal{Y}_{\mathcal{T}j}, \quad (6)$$

where \mathcal{W} is a $|\mathcal{C}_{\mathcal{S}}|$ -dimensional weight vector. The irrelevant categories ($\mathcal{C}_{\mathcal{S}} - \mathcal{C}_{\mathcal{T}}$) will have a much smaller weight than the shared categories. We then assign the weight as 0 if its element \mathcal{W}_c is less than a sufficiently small number (e.g., $10e - 9$). By reducing the weight of irrelevant categories, the shared categories can be emphasized and negative transfer will be mitigated. The weight vector \mathcal{W} is applied in both the source classifier and domain discriminator over the source domain data as shown in the following objective function.

3.8 Overall objective

We combine the three aforementioned loss functions to formalize our objective function:

$$\begin{aligned} & \mathcal{L}(\mathcal{X}_{\mathcal{S}}, \mathcal{X}_{\mathcal{T}}, \mathcal{Y}_{\mathcal{S}}, G_{\mathcal{X}_{\mathcal{S}} \rightarrow \mathcal{X}_{\mathcal{T}}}, G_{\mathcal{X}_{\mathcal{T}} \rightarrow \mathcal{X}_{\mathcal{S}}}) \\ & = \mathcal{L}_{\mathcal{S}}(f(\mathcal{W}(G(\Phi(\mathcal{X}_{\mathcal{S}}))))), \mathcal{Y}_{\mathcal{S}}) + \gamma \mathcal{L}_{\mathcal{A}}(\mathcal{W}(G(\Phi(\mathcal{X}_{\mathcal{S}}))), G(\Phi(\mathcal{X}_{\mathcal{T}}))) \\ & + \beta \mathcal{L}_{Con}(G_{\mathcal{X}_{\mathcal{S}} \rightarrow \mathcal{X}_{\mathcal{T}}}, G_{\mathcal{X}_{\mathcal{T}} \rightarrow \mathcal{X}_{\mathcal{S}}}, \mathcal{W}(G(\Phi(\mathcal{X}_{\mathcal{S}}))), G(\Phi(\mathcal{X}_{\mathcal{T}}))), \end{aligned} \quad (7)$$

where γ and β are tradeoff parameters between different loss functions. Our model ultimately minimizes the difference during the transition from the source domain to target domain and from the target domain to the source domain. Meanwhile, it maximizes the ability to distinguish the two domains.

3.9 Gradients of shared layers

The shared layers consist of the feature extractor G and the feature reconstruction layers. In G , there are two dense layers, a ‘‘Relu’’ activation layer, and a dropout layer. The numbers of units of the dense layer are 1000 and 997, respectively. The rate of the Dropout layer is 0.5. The feature reconstruction layers have a ‘‘Relu’’ activation layer, a dropout layer and a dense layer with the number of units of 1000. The shared layers are jointly optimized by both the source classification loss, adversarial domain loss and feature consistency loss.

Let $\mathcal{F}_{\mathcal{E}}(\cdot, \theta_{\mathcal{E}})$ be the output of the shared encoder with parameters of $\theta_{\mathcal{E}}$. In addition, let $\mathcal{F}_{\mathcal{S}}(\cdot, \theta_{\mathcal{S}})$ be the output of class label classifier with parameters of $\theta_{\mathcal{S}}$, $\mathcal{F}_{\mathcal{A}}(\cdot, \theta_{\mathcal{A}})$ be the output of domain label predictor with parameters of $\theta_{\mathcal{A}}$, and $\mathcal{F}_{Con}(\cdot, \theta_{Con})$ be the output of feature consistency regressor with parameters of θ_{Con} . Therefore, the shared layers are optimized by these three gradients. The parameters in the shared layers are updated in the following equation:

$$\begin{aligned} \theta_{\mathcal{S}} \leftarrow \theta_{\mathcal{S}} - \eta \frac{\partial \mathcal{L}_{\mathcal{S}}}{\partial \theta_{\mathcal{S}}}, \theta_{\mathcal{A}} \leftarrow \theta_{\mathcal{A}} - \eta \tau \theta_{\mathcal{A}} \frac{\partial \mathcal{L}_{\mathcal{A}}}{\partial \theta_{\mathcal{A}}}, \theta_{Con} \leftarrow \theta_{Con} - \eta \frac{\partial \mathcal{L}_{Con}}{\partial \theta_{Con}} \\ \theta_{\mathcal{E}} \leftarrow \theta_{\mathcal{E}} - \eta \left(\frac{\partial \mathcal{L}_{\mathcal{S}}}{\partial \theta_{\mathcal{S}}} + \tau \theta_{\mathcal{A}} \frac{\partial \mathcal{L}_{\mathcal{A}}}{\partial \theta_{\mathcal{A}}} + \frac{\partial \mathcal{L}_{Con}}{\partial \theta_{Con}} \right), \end{aligned} \quad (8)$$

where η is the learning rate and τ is the adaptation factor from gradient reversal layer (GRL) in [4].

3.10 Theoretical Analysis

We now formalize the error bound of our model. *ACL* model is trained with both the labeled source domain and the unlabeled target domain. The error bound of the source domain and the target domain ($\epsilon(h)$) in our model is then formally written as:

$$\epsilon(h) = \epsilon_{\mathcal{X}_S}(h, \mathcal{Y}_S) + \epsilon_{\mathcal{X}_T}(h, \hat{\mathcal{Y}}_T), \quad (9)$$

where $\hat{\mathcal{Y}}_T$ is the predicted label of target domain. The term $\epsilon_{\mathcal{X}_S}(h, \mathcal{Y}_S) = \mathbb{E}_{x \sim \mathcal{X}_S} [|h(x) - \mathcal{Y}_S|]$ and $\epsilon_{\mathcal{X}_T}(h, \hat{\mathcal{Y}}_T) = \mathbb{E}_{x \sim \mathcal{X}_T} [|h(x) - \hat{\mathcal{Y}}_T|]$ denote the expected risk over the source domain and the target domain with respect to the ground truth labels and predicted labels, respectively (where $|\cdot|$ is the L1 norm).

During the training, we expect the error $\epsilon_{\mathcal{X}_T}(h, \hat{\mathcal{Y}}_T)$ to be close to $\epsilon_{\mathcal{X}_T}(h, \mathcal{Y}_T)$, which evaluates the classifier f with true target domain labels. The smaller the difference between these two errors, the better the model performs and more discrepancies of the two domains are reduced. Existing domain adaptation theory shows that the risk in the target domain can be minimized by bounding the source risk and discrepancy between source and target domains [1]). Therefore, the generalization error bound of our model is shown in the following Lemma.

Lemma 1 Let h be a hypothesis in a class H . Then

$$\begin{aligned} \epsilon(h) &= \epsilon_{\mathcal{X}_S}(h, \mathcal{Y}_S) + \epsilon_{\mathcal{X}_T}(h, \hat{\mathcal{Y}}_T) \\ &\leq 2\epsilon_{\mathcal{X}_S}(h) + d_{\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) + C, \end{aligned} \quad (10)$$

where $d_{\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) = 2 \sup_{h, h' \in H} |\epsilon_{\mathcal{X}_S}(h, h') - \epsilon_{\mathcal{X}_T}(h, h')|$ is the \mathcal{H} -divergence of training and test data in the hypothesis space H . $C = \epsilon_{\mathcal{X}_S}(h^*, \mathcal{Y}_S) + \epsilon_{\mathcal{X}_T}(h^*, \mathcal{Y}_T)$ is the adaptability to quantify the error in ideal hypothesis h^* space of training and test data, which should be small and is the optimal hypothesis via minimizing the joint error in Eq. 11.

$$h^* = \arg \min \epsilon_{\mathcal{X}_S}(h, \mathcal{Y}_S) + \epsilon_{\mathcal{X}_T}(h, \mathcal{Y}_T) \quad (11)$$

In Lemma 1, the generalization boundary of our model consists of three terms: training data error, data discrepancy $d_{\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T)$, which is estimated by the disagreement of hypothesis in the space H , and the adaptability C of the ideal joint hypothesis. In *ACL* model, the first term is measured by Eq. 1. The domain discrepancy is assessed by adversarial learning loss and feature consistency loss. Furthermore, *ACL* finds the ideal hypothesis and reduces the training error in each iteration. Hence, our model can find a minimal boundary for two domains. In other words, *ACL* can implicitly minimize the target domain risk, domain discrepancy, and the adaptability of true hypothesis h in terms of the hypothesis space H .

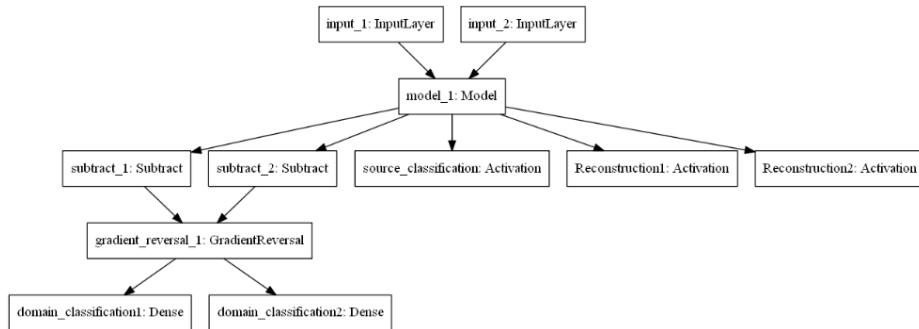


Fig. 3: Layers visualization of our proposed *ACL* model. Two input layers are from source domain data and target domain data, respectively. The intermediate model is the shared layers in Fig. 2. The source classification layer refers to the classifier f , and two reconstruction layers guarantee the feature consistency of two domains. Two subtract layers are used for the domain discriminator. In addition, the gradient reversal layer is used for backpropagation.

4 Experiments

4.1 Implementation details

As aforementioned, the deep features are extracted from the last fully connected layer [15, 17]. One represented feature vector has the size of 1×1000 and is corresponding to one plant image. Therefore, the feature representation of domain herbarium (H) has the size of $320,750 \times 1000$, domain herbarium_photo_associations (A) has the size of $1,816 \times 1000$, domain photo (P) has the size of $4,482 \times 1000$, and domain test (T) has the size of $3,186 \times 1000$. In the experiment, our task is to reduce the error in the target domain (real-world plant images), i.e., photo domain or test domain. Our tasks will focus more on the evaluation of the domain P and domain T. Since the herbarium_photo_associations (A) is important to bridge the map between two domains, we hence include the domain A in the training procedure to form a new source domain, which consists of domain herbarium (H) and domain A. Domain H + A has the size of $322,566 \times 1000$. We then train the model based on these extracted feature vectors. In Tab. 2, H \rightarrow P represents learning knowledge from domain H, which is applied to domain P.

The parameters of *ACL* are first tuned based on the performance of the domain P, while the model is trained with H + A domain. We then apply these parameters to domain T and submit it to the challenge for the evaluation. Our implementation is based on Keras. The parameters settings are $\beta = \gamma = 0.5$, $\tau = 0.31$, learning rate: $\eta = 0.0001$, batch size = 128, the number of iterations is 1000 and the optimizer is Adam. The details of the layers are shown in Fig. 3.

We also compare our results with two domain adaptation methods: DANN [4] and ADDA [14]. In addition, we extracted features from four well-trained models

Table 2: Accuracy (%) on PlantCLEF 2020 dataset for photo domain

Task	A → P	H → P	H+A → P
DANN [4]	1.07	1.85	2.01
ADDA [14]	2.95	3.05	3.43
ResNet50- <i>ACL</i>	2.96	4.83	6.97
InceptionV3- <i>ACL</i>	3.02	5.93	7.95
Inception-Resnet-V2- <i>ACL</i>	3.73	7.07	8.43
NASNetLarge-<i>ACL</i> -\mathcal{W}	3.84	7.92	8.18
NASNetLarge-<i>ACL</i>	5.98	8.64	9.67

(ResNet50 [7], InceptionV3 [13], Inception-Resnet-V2 [12], NASNetLarge [21]), which is trained based on large-scale ImageNet datasets. We then feed these different extracted features into the shared layers and optimize the objective function in Eq. 7.

4.2 Results

The performance of the photo domain is shown in Tab. 2. We report the accuracy of the whole photo domain ($Acc = \sum_{j=1}^{\mathcal{N}_{\mathcal{T}}} (\mathcal{Y}_{\mathcal{T}_j} == \mathcal{Y}_{\mathcal{T}_j}) / \mathcal{N}_{\mathcal{T}} \times 100$), where $\mathcal{Y}_{\mathcal{T}}$ is the predicted label for the target domain. We can observe that the extracted features from NASNetLarge with our *ACL* architecture achieves the highest performance across all three tasks. We observe that two domain adaptation methods have relatively lower performance in all three tasks. One reason is that these two methods have weak feature extractors, and they do not exclude the irrelevant categories in the source domain, which might cause the negative transfer. Moreover, with the increasing of the ImageNet model, we can extract better features from plant images, which lead to the high performance of the NASNetLarge-*ACL* model. In addition, we conduct an ablation study in which we train the best NASNetLarge-*ACL* model without the shared categories selection (NASNetLarge-*ACL* - \mathcal{W}). The results from all three tasks are lower than NASNetLarge-*ACL* model, which indicates the shared categories selection is useful in our model. These experiments demonstrate the efficiency of the *ACL* model in finding the invariant-features of two domains.

In the final stage of the PlantCLEF 2020 Challenge, our solutions are evaluated by the organizers using the test domain data. As shown in Tab. 3, our method achieved mean reciprocal rank (MRR) of 0.032 in the whole test domain, and MRR of 0.016 in the subset of the test domain, and our method places 4th in the contest.

5 Discussion

There are two compelling advantages of the *ACL* model. First, we consider the adversarial consistent learning paradigm, which maintains the domain-invariant features from the source domain to the target domain and vice versa. Secondly,

Table 3: MRR on PlantCLEF 2020 dataset for test domain [5]

Team	Full test set	Sub-set of the test set
ITCR PlantNet	0.180	0.052
Neuron AI	0.121	0.107
UWB	0.039	0.007
LU(ours)	0.032	0.016
Domain	0.031	0.015
To Be	0.028	0.016
SSN	0.008	0.003

we reduce the weight of irrelevant categories in the source domain, which eliminates the negative transfer during the training. Although the performance of our model is better than several baseline methods, the highest accuracy of the photo domain is less than 10%, which illustrates that the transfer learning ability in the real world image is lower. One underlying reason is that PlantCLEF 2020 Challenge has difficult datasets—that there are significant differences between herbarium domain and photo domain, as shown in Fig. 1. Another reason is caused by the weakness of our model since we only train deep features instead of raw images to reduce the computational requirements; some features might be ignored during the training. The performance of the *ACL* model could be improved if we train the architecture with raw images.

6 Conclusion

In this paper, we propose an adversarial consistent learning network on partial domain adaptation termed (*ACL*) to overcome limitations in finding proper shared categories and guaranteeing the feature consistency of two domains. Our model is optimized via minimizing a three-component loss function. As each component of our *ACL* model, explicit domain-invariant features are maintained through such a cross-domain training scheme. Experimental results demonstrate our proposed *ACL* model yields promising results on the PlantCLEF 2020 Challenge.

References

1. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. *Machine Learning* **79**(1-2), 151–175 (2010)
2. Cao, Z., Ma, L., Long, M., Wang, J.: Partial adversarial domain adaptation. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. pp. 135–150 (2018)
3. Chen, J., Wu, X., Duan, L., Gao, S.: Domain adversarial reinforcement learning for partial domain adaptation. *arXiv preprint arXiv:1905.04094* (2019)
4. Ghifary, M., Kleijn, W.B., Zhang, M.: Domain adaptive neural networks for object recognition. In: *Pacific Rim international conference on artificial intelligence*. pp. 898–904. Springer (2014)

5. Goëau, H., Bonnet, P., Joly, A.: Overview of lifeclef plant identification task 2020. In: CLEF working notes 2020, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2020, Thessaloniki, Greece. (2020)
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
8. Joly, A., Goëau, H., Kahl, S., Deneu, B., Servajean, M., Cole, E., Picek, L., Ruiz De Castañeda, R., Lorieul, T., Botella, C., Glotin, H., Champ, J., Vellinga, W.P., Stöter, F.R., Dorso, A., Bonnet, P., Müller, H.: Overview of lifeclef 2020: a system-oriented evaluation of automated species identification and species distribution prediction. In: Proceedings of CLEF 2020, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2020, Thessaloniki, Greece. (2020)
9. Liu, H., Long, M., Wang, J., Jordan, M.: Transferable adversarial training: A general approach to adapting deep classifiers. In: International Conference on Machine Learning. pp. 4013–4022 (2019)
10. Long, M., Cao, Z., Wang, J., Jordan, M.L.: Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems. pp. 1647–1657 (2018)
11. Pan, S.J., Yang, Q., et al.: A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* **22**(10), 1345–1359 (2010)
12. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
13. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
14. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7167–7176 (2017)
15. Zhang, Y., Allem, J.P., Unger, J.B., Cruz, T.B.: Automated identification of hookahs (waterpipes) on instagram: an application in feature extraction using convolutional neural network and support vector machine classification. *Journal of Medical Internet Research* **20**(11), e10513 (2018)
16. Zhang, Y., Davison, B.D.: Modified distribution alignment for domain adaptation with pre-trained inception resnet. *arXiv preprint arXiv:1904.02322* (2019)
17. Zhang, Y., Davison, B.D.: Impact of imagenet model selection on domain adaptation. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops. pp. 173–182 (2020)
18. Zhang, Y., Tang, H., Jia, K., Tan, M.: Domain-symmetric networks for adversarial domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5031–5040 (2019)
19. Zhang, Y., Xie, S., Davison, B.D.: Transductive learning via improved geodesic sampling. In: Proceedings of the 30th British Machine Vision Conference (2019)
20. Zhang, Y., Davison, B.D.: Domain adaptation for object recognition using subspace sampling demons. *Multimedia Tools and Applications* (2020)
21. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8697–8710 (2018)