Unsupervised Domain Adaptation using Feature Aligned Maximum Classifier Discrepancy

Prasanna Reddy Pulakurthi[a], Sohail A. Dianat[a], Majid Rabbani[a], Suya You[b], and Raghuveer M. Rao[b]

[a]Rochester Institute of Technology, Department of Electrical and Microelectronic Engineering, Rochester, New York, United States
[b]DEVCOM Army Research Laboratory, United States

.

ABSTRACT

The maximum classifier discrepancy method has achieved great success in solving unsupervised domain adaptation tasks for image classification in recent years. Its basic structure consists of a feature generator and two classifiers that aim to maximize the classifier discrepancy while minimizing the generator discrepancy of the target samples. This method improves the performance of the existing adversarial training methods by employing task-specific classifiers that remove the ambiguity in classifying the target samples near the class boundaries.

In this paper, we propose a modified network architecture and two training objectives to further boost the performance of the maximum classifier discrepancy method. The first training objective minimizes the feature level discrepancy and forces the generator to generate domain invariant features. This training objective is particularly beneficial when the source and the target domain distributions are vastly different. The second training objective that works at the mini-batch level aims at creating a uniform distribution of the target class predictions by maximizing the entropy of the expectation of the target class predictions. We show through extensive empirical evaluations that the proposed architecture and training objectives significantly improve the performance of the original algorithm. Furthermore, this method also outperforms the state-of-the-art techniques in most unsupervised domain adaptation tasks.

Keywords: Unsupervised domain adaptation, adversarial training, multi-classifier structure, maximum classifier discrepancy, maximum entropy

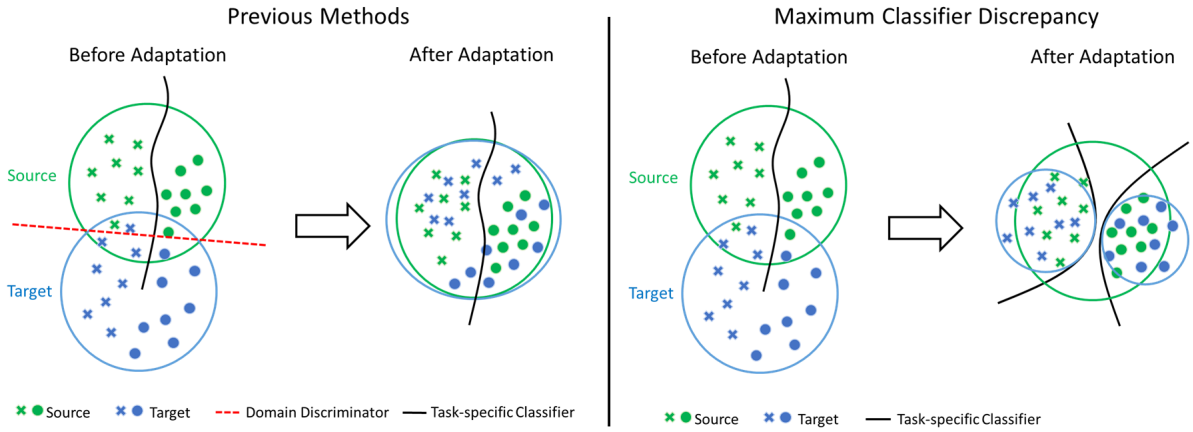
1. INTRODUCTION

With recent developments in deep learning research, Convolutional Neural Networks (CNN) have achieved significant advancements in several computer vision applications, such as image generation,[1,2] segmentation,[3–5] classification,[6–8] object detection,[9–11] and tracking.[12–14] But, these advancements often rely on a large amount of training data. However, in many cases, obtaining a large amount of labeled data (target domain data) is not feasible, but a related yet different training set (source domain data) is readily available. In literature, this problem of translating domain knowledge from rich, labeled source data to unlabeled target data is called Unsupervised Domain Adaptation (UDA).

Ganin et al. (2016)[15] introduced a representation learning method to match the source and target domain features using CNNs and gradient reversal layers to tackle this problem. Tzeng et al.[16] introduced a feature generator trained using Generative Adversarial Networks (GAN) to improve this further. This adversarial training method distinguishes features as either source or target samples using a domain discriminator and then trains the feature generator to generate domain invariant features. Liu and Tuzel (2016)[17] introduced a coupled generative adversarial network consisting of two GANs that are coupled by sharing weights across the

first few layers of the generator and the last few layers of the discriminator. In 2017, with the development of CycleGAN,[18] an image-to-image translation network, it was now possible to align the source and target domain samples at the pixel level in addition to the feature level as proposed by CyCADA.[19]

The drawback of these adversarial training methods is that they try to align the source and target features and do not consider task-specific classification decision boundaries, thus creating ambiguous target features near the class decision boundaries as illustrated in Figure 1a. To overcome this, Saito et al.[20] proposed the Maximum Classifier Discrepancy (MCD) method, which aims to align the source and target features by utilizing two task-specific classifiers. These classifiers act as a discriminator and consider the relationship between class boundaries and target samples, which helps to remove the ambiguity in classifying the target samples near the class boundaries, as it has been illustrated in Ref 20 and shown in Figure 1b. Yang et al. (2021)[21] showed that using multiple classifiers can further improve the MCD method's performance. It is observed that as more classifiers are added, the target classification accuracy increases with an increased algorithm complexity. They conclude that using three classifiers yields the best performance as a trade-off between accuracy and algorithm complexity.



(a) (b)

Figure 1. a) Previous methods performing domain adaptation by adversarial training with the help of a domain discriminator. b) The Maximum Classifier Discrepancy method generates accurate class samples near the class boundaries after adaptation.

In this article, we propose an improved CNN architecture for both the feature generator and task-specific classifiers, as detailed in Section 3, and two training objectives to further enhance the performance of the MCD method. Our method, using two classifiers, not only improves the original MCD performance but also outperforms the other methods that use multiple classifiers. Though using the improved architecture without training objectives performs very well in small domain shifts (e.g., MNIST to USPS domain shift as shown in Figure 2), it faces two problems when dealing with huge domain shifts (e.g., SVHN to MNIST domain shift as shown in Figure 2).

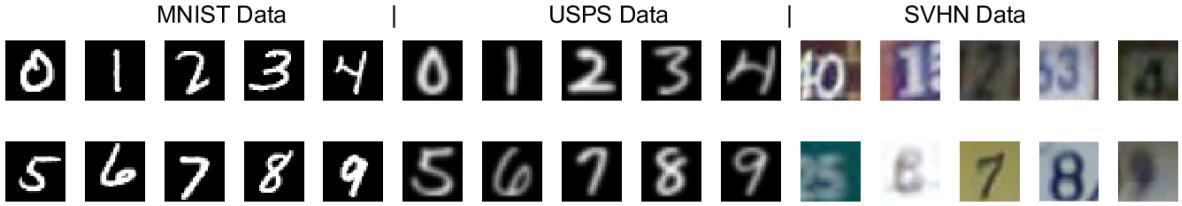
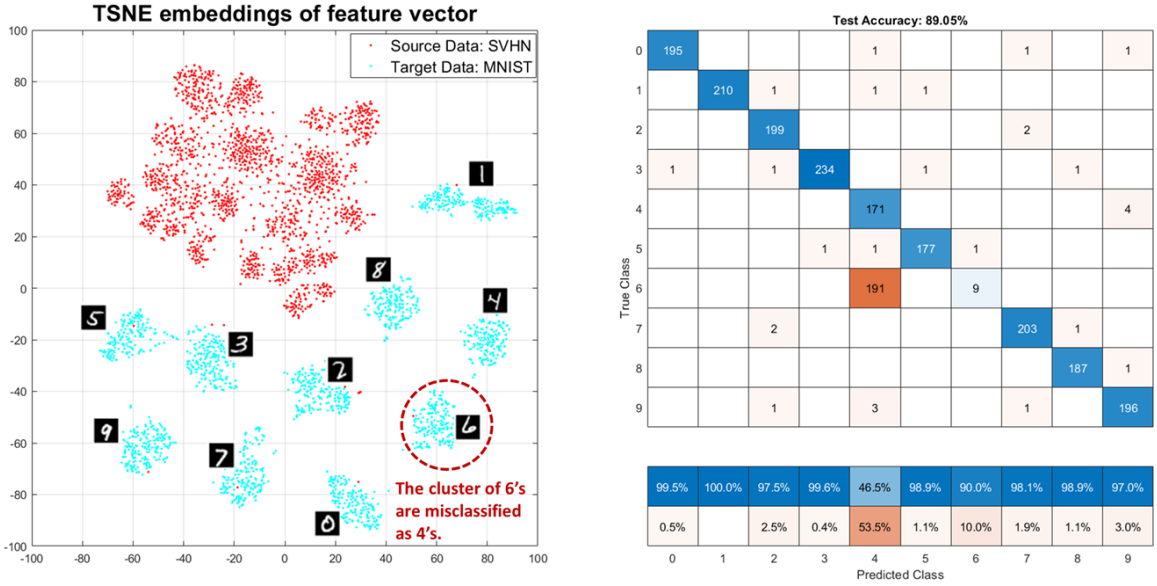


Figure 2. a) Digit images from MNIST dataset,[22] b) USPS dataset,[23] and c) SVHN dataset.[24]

The first problem is that the source and target features fail to align in large domain shifts, as seen in Figure 3a. This is because the MCD method does not enforce any feature space domain matching objectives. The

second problem is that most of the samples from a class might get misclassified as another, as observed from the confusion matrix in Figure 3b, in which all the 6's are misclassified as 4's. To address these problems, we propose two novel loss functions, feature alignment loss, and maximum entropy loss. The first loss function, feature alignment loss, aims to align the source and target domain features by forcing the feature generator to generate domain invariant features. The second loss function, the maximum entropy loss, aims to create a uniform distribution of the output target class predictions in a mini-batch, thus preventing the classifier from missing a particular class prediction.



(a) (b)

Figure 3. a) T-SNE plot of feature vector showing the misaligned source and target distributions and the misclassification of 6's as 4's. b) Confusion matrix of the target predictions for SVHN → MNIST adaptation using the MCD method and proposed architecture.

In summary, the main contributions of this paper are as follows:

1. A modified deep neural network architecture is proposed to improve the target domain performance.

2. Two new loss functions are introduced to overcome the problems when dealing with huge domain shifts, further boosting the target classification performance.

3. The proposed method using two classifiers performs better than methods using three or more classifiers.

2. METHOD

In the unsupervised domain adaptation setting, we have access to rich labeled source domain data and unlabeled target domain data. The source domain data along with the labels are represented as $\{\boldsymbol{X}_s, \boldsymbol{Y}_s\} = \{\boldsymbol{x}_{s_i}, y_{s_i}\}_{i=1}^{N}$ where $\boldsymbol{x}_{s_i} \in R^d$ is the source data sample and the scalar y_{s_i} indicates the class label. The unlabeled target domain is represented by $\boldsymbol{X}_t = \{\boldsymbol{x}_{t_i}\}_{i=1}^{M}$. The source sample $\boldsymbol{x}_s$ belongs to a source distribution $P(\boldsymbol{X}_s)$ and the target sample $\boldsymbol{x}_t$ belongs to a target distribution $P(\boldsymbol{X}_t)$, where $P(\boldsymbol{X}_s) \neq P(\boldsymbol{X}_t)$.

We aim to train a classification model $f_\theta(x)$ to classify the target domain data accurately. The overall network architecture consists of a feature generator G connected to two classifiers $F1$ and $F2$. Each of the classifiers outputs a K-dimensional vector to which a softmax function is applied to generate probability outputs. The classification model is trained with modifications to the method proposed in Ref 20 which consists of three training steps. The first step trains both the feature generator and classifiers to accurately classify the source domain data. In the second step, only the classifiers are trained to maximize the classification discrepancy of the target samples. Finally, in the third step, the feature generator is trained to minimize the classification

discrepancy of the target samples. Our method also trains the network in three training steps with modifications to the training objectives, as detailed below.

Training Step 1: The feature generator and the two classifiers are trained to minimize a combination of three training objectives, the source classification loss, and the two proposed loss functions namely, feature alignment loss and maximum entropy loss, and is written as,

$$Loss_1 = \min_{G,F1,F2}(Loss_{sc} + \lambda_{fa} \times Loss_{fa} + \lambda_h \times Loss_h) \tag{1}$$

Where $Loss_{sc}$ is the source classification loss, $Loss_{fa}$ is the feature alignment loss and $Loss_h$ is the maximum entropy loss. λ_{fa} and λ_h are the Lagrange multipliers. We found through several experiments that the optimal values for these Lagrange multipliers are $\lambda_{fa} = 0.25$ and $\lambda_h = 0.5$. More details on the experiments to determine the influence of λ_{fa} and λ_h on the accuracy is detailed in Appendix A.

The source classification loss ($Loss_{sc}$) given by Equation (2) helps to correctly classify the source domain data samples by minimizing softmax cross-entropy L_{CE} between the network predictions and the ground truth labels.

$$Loss_{sc} = L_{CE}(f_{\theta 1}(\boldsymbol{X}_s), \boldsymbol{Y}_S) + L_{CE}(f_{\theta 2}(\boldsymbol{X}_s), \boldsymbol{Y}_S) \tag{2}$$

The feature alignment loss ($Loss_{fa}$) given by Equation (3) aims to minimize the generator extracted feature discrepancy between the source and target samples. The feature alignment loss is defined as the absolute difference between source and target extracted features.

$$Loss_{fa} = ||(\boldsymbol{G}(\boldsymbol{X}_s) - \boldsymbol{G}(\boldsymbol{X}_t)||_1 \tag{3}$$

Where $\boldsymbol{G}(\boldsymbol{X}_s)$ and $\boldsymbol{G}(\boldsymbol{X}_t)$ are the source and target features extracted by the feature generator and $||.||_1$ is the L_1 norm. We choose the L_1 norm over the L_2 norm as it is more robust to outliers.

The maximum entropy loss ($Loss_h$) given by Equation (4) aims to create a uniform distribution of the target class predictions. This is achieved by maximizing the entropy of the expectation of the target class predictions in a mini-batch.

$$Loss_h = -\frac{1}{2}(H(E_{\boldsymbol{X}_t}(f_{\theta 1}(\boldsymbol{X}_t))) + H(E_{\boldsymbol{X}_t}(f_{\theta 2}(\boldsymbol{X}_t)))) \tag{4}$$

Since maximizing a function is the same as minimizing the negative of that function, we introduce a negative sign in Equation (4). The entropy function $H(.)$ is given by,

$$H(\boldsymbol{p}) = -\sum_{k=1}^{K} p_k \ log_K(p_k) \tag{5}$$

Training Step 2: In this step, the feature generator (G) is fixed, and the two classifiers ($F1$, $F2$) are trained to maximize the discrepancy between their target predictions while making accurate source predictions. This is achieved using the loss function defined in Equation (6).

$$Loss_2 = \min_{F1,F2}(Loss_{sc} - Loss_t) \tag{6}$$

Where $Loss_{sc}$ given in Equation (2) is the classification loss of the source samples, and $Loss_t$ given in Equation (7) is the discrepancy loss between the two classifier's target predictions.

$$Loss_t = d(f_{\theta 1}(\boldsymbol{X}_t), f_{\theta 2}(\boldsymbol{X}_t)) \tag{7}$$

For the discrepancy loss given in Equation (8), we follow Ref. 20 and use the absolute difference between the two classifiers' probability outputs.

$$d(\boldsymbol{p}^1, \boldsymbol{p}^2) = \frac{1}{K} \sum_{k=1}^{K} |p_k^1 - p_k^2| \tag{8}$$

Where $\boldsymbol{p}^1$ and $\boldsymbol{p}^2$ are the probability outputs of the two classifiers, K is the number of classes, and p_k^1 and p_k^2 are the specific values of their k-th class.

Training Step 3: Finally, the feature generator is updated to minimize the discrepancy between the two classifiers as given in Equation (9). It was found that repeating this step four times improves the accuracy as compared to a single iteration as was proposed in Ref. 20.

$$Loss_3 = \min_G d(f_{\theta 1}(X_t), f_{\theta 2}(X_t)) \tag{9}$$

Since the classifiers are fixed, so are the decision boundaries. Therefore, to minimize the prediction discrepancy between these classifiers, the feature generator must extract target features that are consistent with the source extracted features.

3. NETWORK ARCHITECTURE

In this section, we present the modified feature generator and classifier architectures. Garbin et al. (2020)[25] showed success in using the Batch Normalization layer after an activation layer instead of before activation. We develop our deep learning architecture based on this work and observe improved classification performance for domain adaptation. The feature generator and classifier architectures are shown in Figure 4 and Figure 5, respectively. The feature generator contains a deeper architecture than the original MCD method enabling more descriptive features to be extracted. The classifier has multiple fully connected layers allowing for a complex decision boundary to form, thus further improving the target classification accuracy. In the feature generator network, all the convolution filters are of size 3×3, and conv1_64 represents convolution layer 1 with a filter size of 64. The Max-pooling layers use a window of size 2×2 and a stride of 2. In the classifier network, fc1_2048 represents fully-connected layer 1 with 2048 neurons.

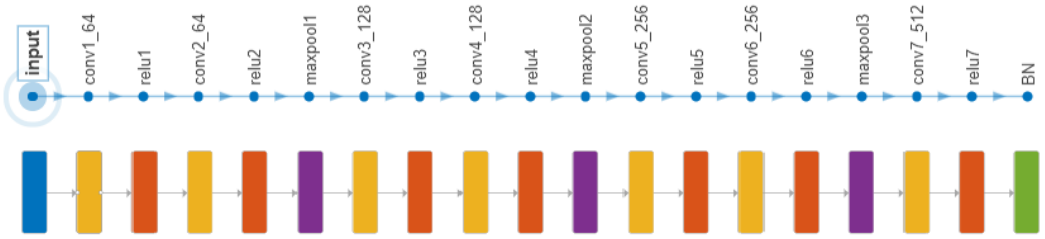


Figure 4. Feature generator architecture.

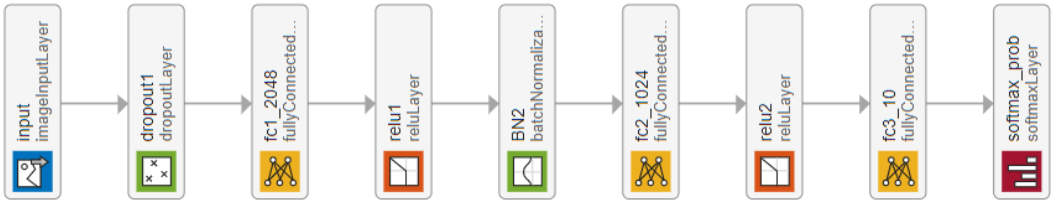


Figure 5. Classifier architecture.

4. RESULTS

Classification Digit Datasets: In this section, we present both the visual performance and target classification accuracy of our proposed method. We compare our method with other state-of-the-art methods in an unsupervised domain adaptation setting. To evaluate our model we used the following digit classification datasets: MNIST,[22] USPS,[23] and Street View House Numbers (SVHN)[24] as shown in Figure 2. All the previous methods of solving unsupervised domain adaptation use these datasets for benchmarking, thus we follow their footsteps to have a meaningful comparison.

a) MNIST dataset has a training set of 60,000 images and a test set of 10,000 images of size 28x28

b) USPS dataset consists of 7,291 training images and 2,007 test images of size 16 x 16 resized to 28x28.

c) SVHN dataset contains 73,257 training, and 26,032 testing color images of size 32x32.

The performance of our method is evaluated on three digit classification domain adaptation scenarios: MNIST → USPS, USPS → MNIST, and SVHN → MNIST. The first two domain shifts are comparatively easy as they do not contain huge domain shifts. However, the third domain shift is challenging and contains large domain distribution divergence because SVHN digit images have a colored background and some extremely blurred images, whereas MNIST contains clear grayscale images. For SVHN → MNIST domain shift, all the MNIST images are resized to 32x32 and are converted to color format. We follow the protocol used in Ref. 20 and Ref. 21, using all the training data.

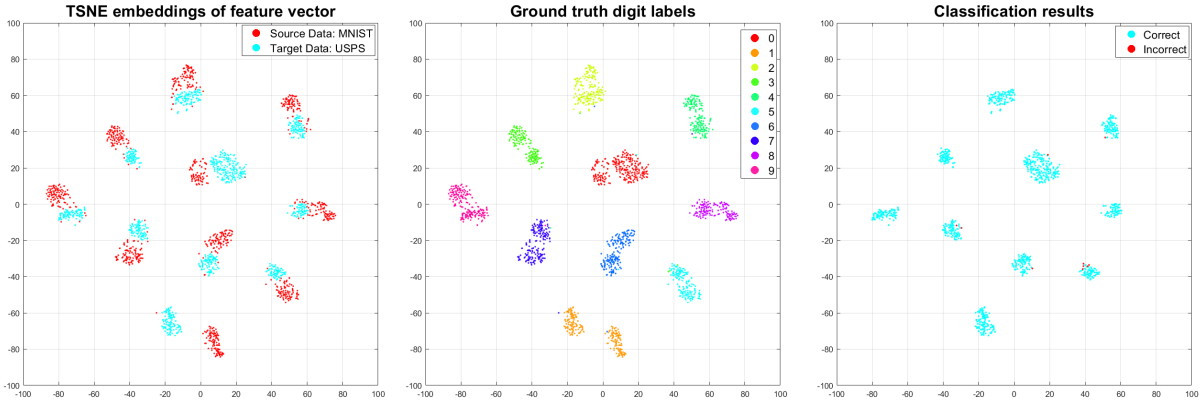
Network and Training: The modified feature generator and classifier network architecture used in all the experiments is presented in Section 3. We used Adam[26] to optimize our model and set the gradient decay factor to 0.5, squared gradient decay factor to 0.999, and learning rate to 0.0002 in all our experiments. The Adam optimizer parameters chosen are widely used in the literature and shown to be effective in a wide range of optimization tasks. To train our model, we used a batch size of 512, as this was the largest batch size we could run on a single NVIDIA RTX 3080 graphic card with 16GB of memory.

Target Classification Results: The target sample classification accuracy of our method is compared to MCD and other methods as shown in Table 1. The proposed method improves the original MCD method's accuracy by 2.22% on MNIST → USPS, and 2.56% on SVHN → MNIST domain shifts. Our method also performs better than the multiple classifier based MCD method[21] by 0.22% on MNIST → USPS, 1.75% on USPS → MNIST and 0.56% on SVHN → MNIST domain shifts.

Table 1. Experimental results of domain adaptation on digit classification domain shifts. Each experiment is repeated five times, and the mean and standard deviation is reported.

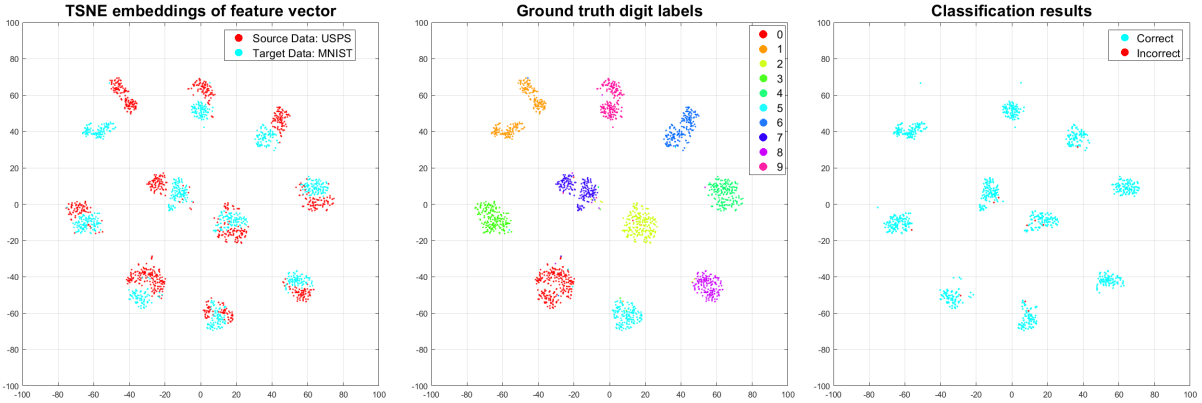
No	Model Name	MNIST → USPS	USPS → MNIST	SVHN → MNIST
1	DANN[15]	85.1	73.0 ± 0.2	71.1
2	ADDA[16]	89.4 ± 0.2	90.1 ± 0.8	76.0 ± 1.8
3	CoGAN[17]	91.2 ± 0.8	89.1 ± 0.8	-
4	CyCADA[19]	95.6 ± 0.2	96.5 ± 0.1	90.4± 0.4
5	MCD[20]	96.5 ± 0.3	-	96.2 ± 0.4
6	MMCD[21]	98.5 ± 0.2	97.0 ± 0.1	98.2 ± 0.1
7	**Proposed Method**	**98.72 ± 0.33**	**98.75 ± 0.12**	**98.76 ± 0.10**

Visual Results: The T-SNE embedding of the feature vectors extracted by the feature generator for MNIST → USPS, USPS → MNIST, and SVHN → MNIST domain shifts are shown in Figure 6, Figure 7, and Figure 8, respectively. Here we can observe clustering of target samples into their corresponding classes, which is consistent with source classification decision boundaries, thus achieving high classification accuracy on the target samples.



(a) (b) (c)

Figure 6. T-SNE plot of the feature vector for MNIST → USPS adaptation. a) Source and target samples. b) Ground truth labels. c) Accurate classifications of target samples.



(a) (b) (c)

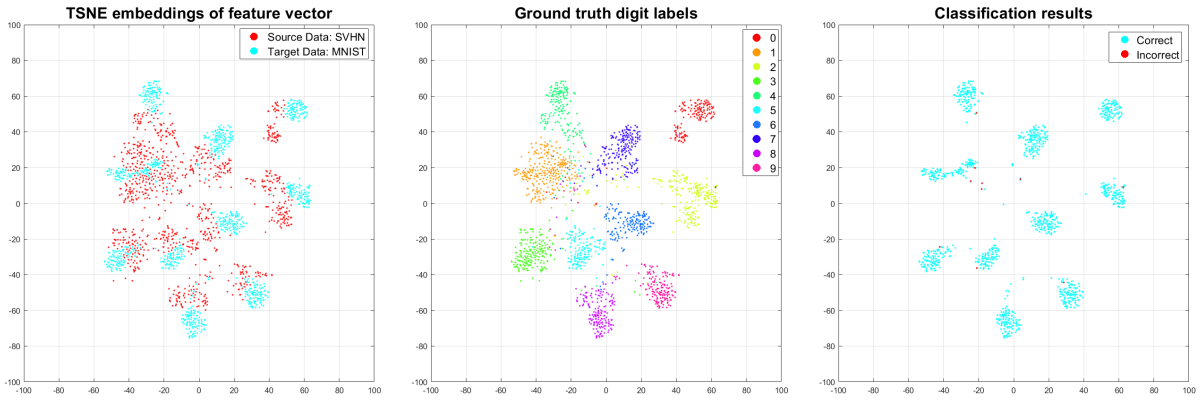
Figure 7. T-SNE plot of the feature vector for USPS → MNIST adaptation. a) Source and target samples. b) Ground truth labels. c) Accurate classifications of target samples.

5. CONCLUSION

This paper presents a modified deep neural network architecture to improve the performance of the maximum classifier discrepancy adversarial training framework. However, this modification has two drawbacks, which are overcome by introducing two novel training objectives: feature alignment loss and maximum entropy loss. The feature alignment loss aims to match the source and target features extracted by the feature generator. The maximum entropy loss seeks to create uniform target class predictions in a mini-batch. Extensive experiments show that the modified network architecture and the two training objectives achieve significant improvement over the previous state-of-the-art domain adaptation methods in the image classification task.

APPENDIX A. ABLATION STUDY

In this section, we study the influence of λ_{fa} and λ_h in equation 1 on the final test accuracy. The experiment is set up by training our method on SVHN to MNIST adaptation for 20 epochs by varying λ_{fa} and λ_h from 0 to 1 in steps of 0.25. A heatmap of the target accuracy is created for different values of λ_{fa} and λ_h, as shown in 9. The first important observation is from the first column of the heat map (i.e., $\lambda_h = 0$), which shows poor target performance in the absence of Maximum Entropy Loss, which highlights the importance of λ_h. For all the values of $\lambda_{fa} \geq 0.25$, best target accuracy is observed when $0.5 < \lambda_h < 0.75$. Our final choice for λ_h is to set $\lambda_h = 0.5$. This choice is because we assume that the target class sample distribution is uniform, which is not always true. Further, using a higher λ_h, in general, might lead to worse performance for other datasets with a



(a) (b) (c)

Figure 8. T-SNE plot of the feature vector for SVHN → MNIST adaptation. a) Source and target samples. b) Ground truth labels. c) Accurate classifications of target samples.

skewed target class distribution. The second important observation is that when $\lambda_h > 0$, $\lambda_{fa} = 0.25$ performs the best, therefore our final choice for the Lagrange multipliers were $\lambda_{fa} = 0.25$ and $\lambda_h = 0.5$.

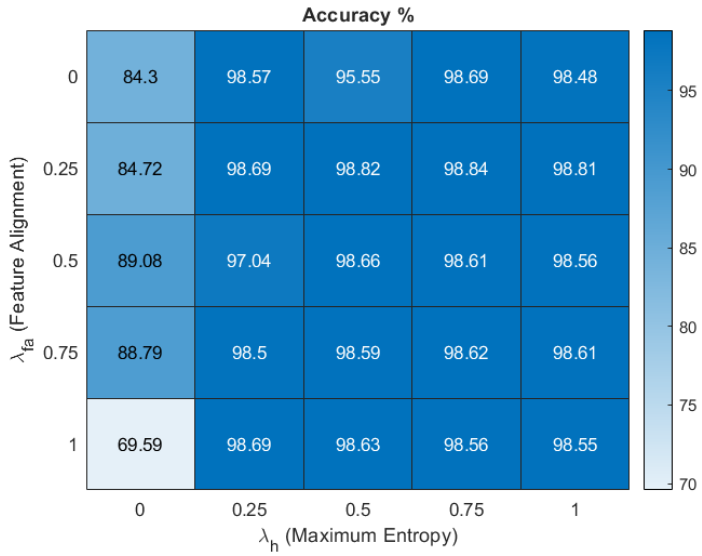


Figure 9. Heatmap of the target accuracy for different values of λ_{fa} and λ_h.

REFERENCES

[1] Lu, Y., Wu, S., Tai, Y.-W., and Tang, C.-K., "Image generation from sketch constraint using contextual gan," in [*Proceedings of the European conference on computer vision (ECCV)*], 205–220 (2018).

[2] Zhou, K., Yang, Y., Hospedales, T., and Xiang, T., "Deep domain-adversarial image generation for domain generalisation," in [*Proceedings of the AAAI Conference on Artificial Intelligence*], **34**(07), 13025–13032 (2020).

[3] Dolz, J., Gopinath, K., Yuan, J., Lombaert, H., Desrosiers, C., and Ayed, I. B., "Hyperdense-net: a hyper-densely connected cnn for multi-modal image segmentation," *IEEE transactions on medical imaging* **38**(5), 1116–1126 (2018).

[4] Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H., "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587* (2017).

[5] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H., "Encoder-decoder with atrous separable convolution for semantic image segmentation," in [*Proceedings of the European conference on computer vision (ECCV)*], 801–818 (2018).

[6] Sun, Y., Xue, B., Zhang, M., Yen, G. G., and Lv, J., "Automatically designing cnn architectures using the genetic algorithm for image classification," *IEEE transactions on cybernetics* **50**(9), 3840–3854 (2020).

[7] Afzal, M. Z., Kölsch, A., Ahmed, S., and Liwicki, M., "Cutting the error by half: Investigation of very deep cnn and advanced training strategies for document image classification," in [*2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*], **1**, 883–888, IEEE (2017).

[8] Lee, H. and Kwon, H., "Going deeper with contextual cnn for hyperspectral image classification," *IEEE Transactions on Image Processing* **26**(10), 4843–4855 (2017).

[9] Hung, J. and Carpenter, A., "Applying faster r-cnn for object detection on malaria images," in [*Proceedings of the IEEE conference on computer vision and pattern recognition workshops*], 56–61 (2017).

[10] Chen, Z., Zhang, J., Ding, R., and Marculescu, D., "Vip: Virtual pooling for accelerating cnn-based image classification and object detection," in [*Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*], 1180–1189 (2020).

[11] Erhan, D., Szegedy, C., Toshev, A., and Anguelov, D., "Scalable object detection using deep neural networks," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 2147–2154 (2014).

[12] He, K., Gkioxari, G., Dollár, P., and Girshick, R., "Mask r-cnn," in [*Proceedings of the IEEE international conference on computer vision*], 2961–2969 (2017).

[13] Plastiras, G., Kyrkou, C., and Theocharides, T., "Efficient convnet-based object detection for unmanned aerial vehicles by selective tile processing," in [*Proceedings of the 12th International Conference on Distributed Smart Cameras*], 1–6 (2018).

[14] Ribera, J., Guera, D., Chen, Y., and Delp, E. J., "Locating objects without bounding boxes," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 6479–6489 (2019).

[15] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V., "Domain-adversarial training of neural networks," *The journal of machine learning research* **17**(1), 2096–2030 (2016).

[16] Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T., "Adversarial discriminative domain adaptation," *CoRR* **abs/1702.05464** (2017).

[17] Liu, M. and Tuzel, O., "Coupled generative adversarial networks," *CoRR* **abs/1606.07536** (2016).

[18] Zhu, J., Park, T., Isola, P., and Efros, A. A., "Unpaired image-to-image translation using cycle-consistent adversarial networks," *CoRR* **abs/1703.10593** (2017).

[19] Hoffman, J., Tzeng, E., Park, T., Zhu, J., Isola, P., Saenko, K., Efros, A. A., and Darrell, T., "Cycada: Cycle-consistent adversarial domain adaptation," *CoRR* **abs/1711.03213** (2017).

[20] Saito, K., Watanabe, K., Ushiku, Y., and Harada, T., "Maximum classifier discrepancy for unsupervised domain adaptation," *CoRR* **abs/1712.02560** (2017).

[21] Yang, Y., Kim, T., and Wang, G., "Multiple classifiers based maximum classifier discrepancy for unsupervised domain adaptation," *CoRR* **abs/2108.00610** (2021).

[22] LeCun, Y. and Cortes, C., "MNIST handwritten digit database," (2010).

[23] Hull, J., "A database for handwritten text recognition research," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**(5), 550–554 (1994).

[24] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y., "Reading digits in natural images with unsupervised feature learning," in [*NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*], (2011).

[25] Garbin, C., Zhu, X., and Marques, O., "Dropout vs. batch normalization: an empirical study of their impact to deep learning," *Multimedia Tools and Applications* **79**(19), 12777–12815 (2020).

[26] Kingma, D. P. and Ba, J., "Adam: A method for stochastic optimization," in [*3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*], Bengio, Y. and LeCun, Y., eds. (2015).