https://doi.org/10.2352/EI.2024.36.8.IMAGE-238

Adapt to Distill or Distill to Adapt

Georgi Thomas and Andreas Savakis; Rochester Institute of Technology; Rochester, NY, USA.

Abstract

Domain Adaptation (DA) techniques aim to overcome the domain shift between a source domain used for training and a target domain used for testing. In recent years, vision transformers have emerged as a preferred alternative to Convolutional Neural Networks (CNNs) for various computer vision tasks. When used as backbones for DA, these attention-based architectures have been found to be more powerful than standard ResNet backbones. However, vision transformers require a larger computational overhead due to their model size. In this paper, we demonstrate the superiority of attention-based architectures for domain generalization and source-free unsupervised domain adaptation. We further improve the performance of ResNet-based unsupervised DA models using knowledge distillation from a larger teacher model to the student ResNet model. We explore the efficacy of two frameworks and answer the question: is it better to distill and then adapt or to adapt and then distill? Our experiments on two popular datasets show that adapt-to-distill is the preferred approach.

Introduction

Domain adaptation methods using deep learning aim to mitigate the effects of the domain shift between the source domain where training takes place and the target domain used for testing. We consider the task of unsupervised closed-set Domain Adaptation (DA), where the target domain is unlabelled and the same classes are present in both the source and target domains. The de-facto backbone for domain adaptation methods has been ResNet [11]. However, vision transformers have emerged as popular architectures for computer vision that are replacing Convolutional Neural Networks (CNNs). Transformer architectures achieve state-of-the-art results across many tasks, including domain generalization and unsupervised domain adaptation where there are no labels in a target domain. The self-attention mechanism in transformer architectures has proven to be effective in extracting robust features across different parts of the image. However, the drawback of vision transformers is that they require a much larger computational overhead and memory footprint compared to CNNs.

In this paper, we explore leveraging these attention-based models to help the CNN model better adapt to the target domain. We consider various self-attention models as teachers and perform knowledge distillation to a smaller ResNet student model. In this paper, we enhance the performance of a ResNet-50 student model throught knowledge distillation by different attention-based models for domain generalization and adaptation tasks. Our work makes the following contributions:

- We use knowledge distillation to leverage the feature extraction power of a larger attention-based model as a teacher and improve the adaptation capacity of a smaller ResNet-50 student model.
- We explore transformers and attention-based convolutional models to serve as teachers and help the student model adapt to the target domain.
- We compare two different adaptation-distillation methods for source-free unsupervised DA and determine that it is better to adapt the teacher model first and then distill to the smaller student model, as opposed to distilling first and then adapting the smaller model.

Related Work

Source-Free Domain Adaptation

The domain shift or domain gap occurs when there is a distribution shift between the source domain data used for training and the target domain data used for testing. The domain gap significantly reduces the performance of source-trained models during deployment in the target domain. Domain adaptation methods try to align the features of the source-trained model to the target domain and mitigate the domain shift. Multiple DA approaches exist, such as adversarial and source-free adaptation. Adversarial methods use the source data in the adaptation process to adversarially align the features in the source and target domains. Source-free DA methods do not require the source data during adaptation, and use information from the source-trained model to align the features of the source and target domains. A popular source-free DA method is Source HypOthesis Transfer (SHOT) [9], which combines pseudo labels generated with the deep cluster method and information maximization.

Attention-based Architectures

The success of transformers for natural language processing (NLP) has popularized the attention mechanism leading to vision transformer architectures. The attention mechanism allows the model to extract better global features by capturing long-range dependencies between words in a sentence. Inspired by transformers for language, Vision Transformers (ViT) [13] were introduced for various image tasks. ViT splits an image into patches, to mimic words in a sentence, before processing each patch though the transformer encoder. However, ViT is quadratic in complexity and doesn't properly aggregate features across different parts of a patch. The Shifted Window Transformer (SWIN) [10] was designed to overcome some of the issues with ViT by splitting the image into smaller patches. By doing so, SWIN can aggregate features across the smaller patches and extract long-range global features across the image.

Following the emergence of transformer architectures in computer vision, ConvNeXt [12] was proposed as a convolutional network alternative to vision transformers. ConvNext modernized the ResNet architecture by mimicking the attention mechanism.

By patchifying the convolutional layer and using an inverted bottleneck, ConvNeXt showed that convolutional models can perform comparably to the vision transformer architectures.

Transformer models have demonstrated better generalization properties than CNNs under domain shift [5] and are an attractive alternative as a powerful backbone for source-free domain adaptation [5], [17]. However, they are larger and computationally more expensive than CNNs, which is not well-suited for edge computing or mobile platforms. In this paper, we seek to leverage the power and robustness of transformer architectures by distilling their knowledge to a smaller ResNet-50 model for more effective adaptation to the target domain.

Knowledge Distillation in Domain Adaptation

Knowledge distillation was introduced in [6] for model compression by transferring knowledge from a larger teacher model to a smaller student model. In domain adaptation, the work in [7] proposed knowledge distillation as a way to replace source training for an adapting model. This approach harnesses the generalization capability of a ViT model and distills its knowledge of the target domain to the adapting ResNet model. The ResNet model adapts to the target domain by exploiting the information maximization loss to diversify its outputs. In a different approach, [8] first adapted a larger model to the target domain using unsupervised domain adaptation and then distilled the knowledge to a smaller model. They showed that the smaller distilled model can perform better than the larger adapted model.

In this paper, we consider two approaches for unsupervised domain adaptation with knowledge distillation presented in the next section. In both cases, we begin with source-training of a larger transformer-based backbone and perform knowledge distillation to a smaller CNN backbone for deployment to the target domain.

Methodology

We investigate two frameworks for domain adaptation with knowledge distillation. First in the *distill to adapt* framework illustrated in Figure 1, we distill a larger source-trained backbone to a smaller CNN model that is subsequently adapted to the target domain. Second, in the *adapt to distill* framework illustrated in Figure 2, we adapt the larger model to the target domain and then distill it to the smaller CNN model.

In a closed-set unsupervised DA setting, training the source model is performed with n_s labeled samples $\{(x_s, y_s) \in (X_s, Y_s)\}$ from the source domain D_s. Adaptation is based on n_t unlabeled samples $\{x_t \in X_t\}$ from the target domain D_t. The domain adaptation task involves determining a mapping $(f_t : X_t \rightarrow Y_t)$ to the corresponding labels $\{y_t \in Y_t\}$ for the target domain. Under the closed-set setting, we assume that the same classes are present in the source and target domains, namely that $C_t = C_s$.

Source Training

The models for the teacher and student networks include a feature extractor backbone and a classification head. The teacher and student networks are trained on the source data $\{(x_s, y_s) \in (X_s, Y_s)\}$ in the source domain D_s. The feature extractors for the teacher and student networks are denoted as G_s and g_s, respectively. Likewise, the classification head for the teacher and student as H_s and h_t respectively. The source teacher predictions are F_s, where $F_s = H_s(G_s)$ and the source student inferences are f_s, where $f_s = h_s(g_s)$. The feature extraction module includes a feature extraction backbone and a batch normalization layer. The classifier hypothesis module includes a weight normalization layer.

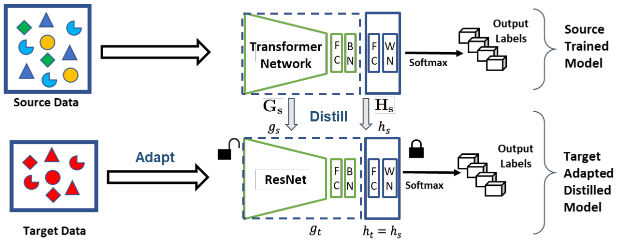

Figure 1. *Distill to adapt framework. A larger source-trained backbone is distilled to a smaller ResNet model, which is adapted to the target domain.*

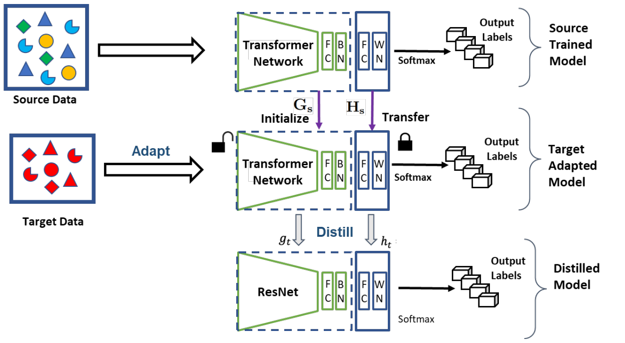

Figure 2. *Adapt to distill framework. A larger source-trained model is adapted to the target domain and after adaptation it is distilled to the smaller ResNet model.*

Using the features extracted from the image, the classifier head returns K logits, where $K = C_t = C_s$ is the total number of classes within the dataset. We train on the source data by minimizing the cross-entropy loss using label smoothing for the source training procedure. Label smoothing increases the model's ability to generalize across multiple classes by improving the clustering of samples. Smoothed labels help to decrease the gaps between the predictions and prevent the model from being overconfident by softening the one-hot encoding.

We use q as the one-hot encoding of the output, where q is defined to be '1' for the intended class and '0' for any other class. For the k-th element, the true label is q_k and the smoothed label is q_k^{ls}, defined as:

$$q_k^{ls} = (1-\alpha)q_k + \alpha/k \tag{1}$$

where α is the label smoothing parameter that is set to 0.1. We use the softmax probability

$$\delta_k(a) = \frac{exp(a_k)}{\sum_i exp(a_i)} \tag{2}$$

where $\delta_k(a)$ denotes the k-th element in the softmax output of a K-dimensional vector a. We incorporate this into the cross-entropy loss function which becomes

$$\mathscr{L}_{src}(f_s; X_s, Y_s) = -\mathbf{E}_{(x_s, y_s) \in \{X_s, Y_s\}} \sum_{k=1}^{K} q_k^{ls} \; log \; \delta_k(f_s(x_s)) \tag{3}$$

Clustering for Pseudo-Labels

To generate pseudo-labels for target adaptation, we first determine the initial centroids, $c_k^{(0)}$, using the softmax output of the target samples,

$$c_k^{(0)} = \frac{\sum_{x_t \in X_t} \delta_k(p_t(x_t))\ (F_t(x_t))}{\sum_{x_t \in X_t} \delta_k(F_t(x_t))} \tag{4}$$

where p_t describes the previously learned target hypothesis and F_t are the current predictions. We use the cosine distance function and minimize the distance between samples where $D(a,b)$ is the cosine distance function and a and b are the two samples.

$$\hat{y}_t^{(0)} = arg_k min(D(F_t(x_t), c_k^{(0)}) \tag{5}$$

After extracting the initial pseudo-labels, the cluster centers are recomputed as follows.

$$c_k^{(1)} = \frac{\sum_{x_t \in X_t} \mathbf{1}(\hat{y}_t = k)\ (F_t(x_t))}{\sum_{x_t \in X_t} \mathbf{1}(\hat{y}_t = k)} \tag{6}$$

The final pseudo-labels are then computed with the updated cluster centers using

$$\hat{y}_t^{(1)} = arg_k min(D(F_t(x_t),\ c_k^{(1)}) \tag{7}$$

where the $y_t^{(1)}$ are the pseudo-labels extracted from the input target data X_t.

Adaptation Objective Function

During the adaptation process, the classifier head is frozen to improve the feature extraction head. As shown below, we use the pseudo-labels from the clustering algorithm to minimize the cross-entropy loss, $\mathscr{L}_{Tce}$, for the adapting model using the target samples.

$$\mathscr{L}_{Tce}(F_s;X_s,Y_s) = -\mathbf{E}_{(x_t,\hat{y}_t)\in\{X_t,\hat{Y}_t\}} \sum_{k=1}^{K} \mathbf{1}_{[k=\hat{y}_t]}\ log\ \delta_k(F_t(x_t)) \tag{8}$$

Additionally, following [14], we adopt the Information Maximization (IM) loss which aims to increase the diversity among inferences of the model during adaptation. Information Maximization(IM) is a combination of entropy loss $\mathscr{L}_{ent}$ [15] and diversity loss $\mathscr{L}_{div}$ [9]. We define $\mathscr{L}_{ent}$ and $\mathscr{L}_{div}$ as follows:

$$\mathscr{L}_{ent}(f_t;X_t) = -\mathbf{E}_{(x_t)\in\{X_t\}} \sum_{k=1}^{K} \delta_k(f_t(x_t))\ log\ \delta_k(f_t(x_t)) \tag{9}$$

$$\mathscr{L}_{div}(f_t;X_t) = \sum_{k=1}^{K} q_k\ log\ (q_k) \tag{10}$$

We denote f_t to be some target model adapting to the target data and q_k as the mean softmax output of the target data seen by the model. The equal diversity loss $\mathscr{L}_{div}$ aims to make network predictions diverse for all classes to prevent similar one-hot encodings of the observed target data. We then define IM loss as $\mathscr{L}_{IM}$, where

$$\mathscr{L}_{IM}(f_t;X_t) = \mathscr{L}_{ent} + \mathscr{L}_{div} \tag{11}$$

Using the model's cross-entropy loss, $\mathscr{L}_{Tce}$, with the IM loss, $\mathscr{L}_{IM}$, our final adaptation objective function becomes,

$$\mathscr{L}_{F_t} = \mathscr{L}_{Tce} + \mathscr{L}_{IM} \tag{12}$$

Distillation Step

In order to distill the knowledge from the larger attention-based architecture to the ResNet-50 model, we use the Kullback-Leibler loss or $\mathscr{L}_{kl}$ [16] defined as follows.

$$\mathscr{L}_{kl}(F_t(x_t)||f_t(x_t)) = \sum_{x_t \in X_t} F_t(x_t)\ log\left(\frac{F_t(x_t)}{f_t(x_t)}\right) \tag{13}$$

We treat the labels from the teacher model as strong labels and use the following consistency loss,

$$\mathscr{L}_{KD}(f_t;X_t,F_t) = \mathbf{E}_{x_t \in X_t}\ \mathscr{L}_{kl}\ (F_t(x_t)\ ||\ f_t(x_t)) \tag{14}$$

The issue with using this consistency loss is that KD is often used in a supervised setting. However, with unsupervised DA, the teacher outputs for some target instances may be inaccurate. Inspired by [7], we use Adaptive Label Smoothing (AdaLS) to generate a revised output of $\hat{p}$. We have the teacher F_t revise the output p with the top-r values. We consider T_p^r as the set of the top-r labels of classes in the original output p.

$$\hat{p}(r) = \begin{cases} p_i, & i \in T_p^r \\ (1 - \sum_{j \in T_p^r} p_j)\ /\ (K-r), & otherwise \end{cases} \tag{15}$$

We empirically select $r = 1$ to choose the top class and smooth out the remaining labels. We use these refined pseudo-labels to reduce the noisiness of the labels extracted from the teacher and impose a uniform distribution on the labels similar to label smoothing. Smoothing to highlight the most confident classes allows the student to learn from the samples that the teacher is confident about rather than learning from noisy labels that may occur with the domain shift.

Experimental Setup

Adapt-To-Distill Models

For the teacher networks in the two configurations shown in Figures 1 and 2, we consider transformer architectures ViT and SWIN as the backbone. We also utilize the ConvNeXt convolutional architecture. For the student model in both frameworks, we use the popular ResNet-50 backbone for a direct comparison to other domain adaptation methods based on ResNet-50. For both the teacher and the student models, we use an image size of 224 × 224. A comparison of the number of parameters in each of the backbones is presented in Table 1.

Table 1. Information about each backbone architecture used.

Model	Image Size	Parameters
ResNet	224×224	23M
ConvNeXt	224×224	89M
ViT	224×224	86M
SWIN	224×224	88M

Datasets

In order to benchmark and analyze the Adapt-To-Distill and Distill-to-Adapt frameworks, we conduct our experiments on two popular domain adaptation datasets: OfficeHome and

DomainNet-126. OfficeHome [1] is a domain adaptation dataset containing 15,500 images and 65 object classes from 4 different domains: Art (**Ar**), Clipart (**Cl**), Real World (**Rw**), and Product (**Pr**). Some samples of the images from the dataset are shown below in Figure 3.

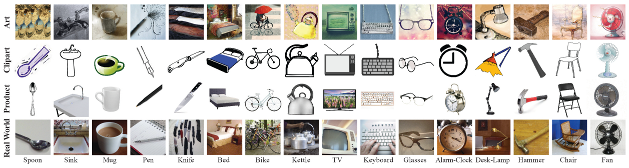

Figure 3. *Sample Images from OfficeHome [1] Dataset*

DomainNet-126 [2] is a subsection of the DomainNet dataset. The DomainNet dataset consists of 600,000 images across 6 domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. Following [3], we use 126 classes from 4 of the 6 domains: Real (**R**), Clipart (**C**), Painting (**P**), and Sketch (**S**) for evaluation. Shown in Figure 4 are 16 of 126 classes across those 4 domains.

Figure 4. *Sample Images from DomainNet [2] Dataset*

Experimental Setup

In order to keep consistency across all the models, the image size is kept at 224 × 224 pixels. We use PyTorch and Timm (PyTorch Image Model Library) [4] to load the models for analysis, training, and testing. We use a stochastic gradient descent (SGD) optimizer for the teacher and student and use a learning rate of `1e-3` for the teacher model and student model. The layers after the feature extraction backbone have a learning rate of 10 times the learning rate of the backbone. We run the model for source training and target adaptation for 20 epochs each.

To understand the difference between the adaptation with distillation frameworks in Figures 1 and 2, we compare two adaptation recipes: adapt-to-distill and distill-to-adapt. We first test the adapt-to-distill method, where we adapt the attention-based model using SHOT adaptation with different backbones. The adapted teacher model is then distilled to the ResNet-50 model. For the distill-to-adapt experiments, we distill the source-trained teacher's knowledge to the ResNet-50 model and then perform adaptation using SHOT.

Results and Discussion

In this section, we present the results of our experiments using different backbones for knowledge distillation and unsupervised domain adaptation. We evaluate the efficacy of different teacher backbones relative to the performance of the student model. The primary metric that we use to compare the models is classification accuracy. This metric is calculated by dividing the total number of correct predictions by the total number of predictions of the model. In our results, we report the "mean percent accuracy" as the classification accuracy.

Tables 2 and 3 present the results for the generalization and adaptation performance on the OfficeHome dataset. For the generalization results in Table 2, the model is trained on the source data and then tested on the target data without any adaptation. For the results in Table 3, we use the SHOT [9] framework with different backbones adapt the source-trained model to the target domain and test the adapted model on the target data. For brevity, the detailed results in the tables show the performance from Art to the other domains, and for overall performance the average accuracy for all domain transfer pairs is reported.

Table 2. Generalization performance (mPA) with different backbones on the OfficeHome dataset.

Backbone	Ar → Cl	Ar → Pr	Ar → Rw	Avg.
ResNet-50	44.5	65.6	74.1	60.0
ConvNeXt	72.9	86.0	89.3	82.0
ViT	50.5	82.5	86.8	74.0
SWIN	70.1	85.2	89.0	81.6

Table 3. Adaptation performance (mPA) using SHOT with different backbones on the OfficeHome dataset.

Backbone	Ar → Cl	Ar → Pr	Ar → Rw	Avg.
ResNet-50	48.9	72.0	75.5	69.2
ConvNeXt	79.2	92.4	92.2	87.7
ViT	69.8	89.6	90.0	83.4
SWIN	76.2	91.5	91.7	87.1

As expected, the attention-based architectures outperform the smaller ResNet-50 model. Comparing the average accuracy across both tables shows an interesting trend: the generalization capability of the attention-based models outperforms the adaptive capabilities of the ResNet-50 model.

Table 4. Generalization performance (mPA) with different backbones on the DomainNet-126 dataset.

Backbone	C → P	C → R	C → S	Avg.
ResNet-50	47.7	61.3	48.8	56.4
ConvNeXt	74.1	83.4	72.7	77.0
ViT	70.8	80.7	67.5	72.3
SWIN	74.2	83.7	71.6	76.7

Table 5. Adaptation performance (mPA) using SHOT with different backbones on the DomainNet-126 dataset.

Backbone	C → P	C → R	C → S	Avg.
ResNet-50	61.2	77.4	60.0	67.4
ConvNeXt	79.2	90.1	76.5	81.6
ViT	76.3	86.9	71.9	78.0
SWIN	79.8	90.1	75.0	81.7

Tables 4 and 5 show the generalization and adaptation performance for different backbones on the DomainNet-126 dataset. Again, we observe the same pattern where the generalization scores of the attention-based models surpass the performance of the adapted ResNet-50 model. From these results, we can conclude that the generalization capabilities of attention-based architectures are much better than the adaptive capabilities of the

ResNet-50 model. Therefore, we leverage these attention-based architectures with knowledge distillation to improve the ResNet-50 performance in the target domain.

The following experiments determine whether it is better to distill-to-adapt or adapt-to-distill. Using the frameworks shown in Figures 1 and 2, SHOT adaptation with different backbones was performed before or after distillation from the larger teacher network to the smaller student ResNet-50. The results are reported in Tables 6 and 7 for OfficeHome and DomainNet-126 respectively. The results are consistent for both datasets and show that Adapt-to-Distill gives the best results for all of the teacher backbones considered. we consistently find that ConvNext and SWIN are the best performing.

Table 6. Results (mPA) using SHOT with knowledge distillation and various teacher backbones on the OfficeHome Dataset.

Teacher	Distill-to-Adapt	Adapt-to-Distill
ViT	74.8	80.0
SWIN	76.2	82.4
ConvNeXt	76.2	82.3

Table 7. Results (mPA) using SHOT with knowledge distillation and various teacher backbones on the DomainNet-126 dataset.

Teacher	Distill-to-Adapt	Adapt-to-Distill
ViT	73.5	78.0
SWIN	74.3	80.1
ConvNeXt	74.5	80.1

Conclusion

In this work, we utilize knowledge distillation as an effective method to improve the performance of source-free DA using SHOT with a ResNet-50 backbone. Within the SHOT source-free DA framework, we use ConvNeXt, SWIN, and ViT backbones to serve as the teachers for the ResNet-50 student adaptation to the target domain. Our results show that adapting the teacher model and then distilling its knowledge to the student model is a more effective method for domain adaptation. The attention-based models generalize better on the target data, and can better adapt to the target data. By leveraging the generalization and adaptive capabilities of these architectures, the ResNet-50 model adapts to the target domain better than if it had adapted on its own.

Acknowledgements

This research was partly supported by the Air Force Office of Scientific Research (AFOSR) under SBIR grant FA9550-22-P-0009 with Intelligent Fusion Technology, AFOSR grant FA9550-20-1-0039 and the Empire State Development's Division of Science, Technology and Innovation through the University of Rochester Center of Excellence in Data Science. The authors would like to thank RIT Research Computing for making computing resources available for experimentation.

References

[1] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. IEEE Conference on Computer Vision and Pattern Recognition, Proc. pg. 5018. (2017)

[2] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. IEEE/CVF international conference on computer vision, Proc. pg. 1406. (2019).

[3] Jian Liang, Dapeng Hu, and Jiashi Feng. "Domain adaptation with auxiliary target domain-oriented classifier." IEEE/CVF conference on computer vision and pattern recognition, Proc. pg. 16632. (2021).

[4] Ross Wightman. "Pytorch Image Models." https://github.com/huggingface/pytorch-image-models. (2019).

[5] Rajat Sahay, Georgi Thomas, Chowdhury Sadman Jahan, Mihir Manjrekar, Dan Popp, and Andreas Savakis. On the Importance of Attention and Augmentations for Hypothesis Transfer in Domain Adaptation and Generalization. Sensors, 20, 8409. (2023).

[6] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531. (2015).

[7] Jian Liang, Dapeng Hu, Jiashi Feng, and Ran He. Dine: Domain adaptation from single and multiple black-box predictors. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Proc. pg. 8003. (2022).

[8] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Proc. pg. 295. (2022).

[9] Jian Liang, Dapeng Hu, and Jiashi Feng. "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation." International Conference on Machine Learning, pg. 6028. (2020).

[10] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. IEEE/CVF International Conference on Computer Vision. Proc. pg. 10012. (2021).

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. Computer Vision–ECCV 2016: 14th European Conference, Proc. pg. 630. (2016).

[12] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. IEEE/CVF Conference on Computer Vision and Pattern Recognition. Proc. pg. 11976. (2022).

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv preprint arXiv:2010.11929 (2020).

[14] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning Discrete Representations via Information Maximizing Self-Augmented Training. International Conference on Machine Learning. Proc. pg. 1558. (2017).

[15] Yves Grandvalet, and Yoshua Bengio. Semi-supervised Learning by Entropy Minimization. Advances in Neural Information Processing Systems 17. (2004).

[16] Solomon Kullback, and Richard A. Leibler. On information and sufficiency. The Annals of Mathematical Statistics 22, 79 (1951).

[17] Abu Md Niamul Taufique, Chowdhury Sadman Jahan, and Andreas Savakis. ConDA: Continual unsupervised domain adaptation. IEEE Trans. Artificial Intelligence. (2023).

Author Biography

Georgi Thomas received his BS and MS degrees in computer en-

gineering from Rochester Institute of Technology (2023) and is currently with Boeing. His research interests include domain adaptation, novel deep learning architectures and creating efficient deep learning models.

Andreas Savakis is a professor of computer engineering and director of the Center for Human-aware AI (CHAI) at Rochester Institute of Technology. His research interests include computer vision, deep learning, domain adaptation, robust and efficient learning and human pose estimation. He is SPIE Fellow.