# Deep Active Transfer Learning for Image Recognition

Ankita Singh
*Department of Computer Science*
*Florida State University*

Shayok Chakraborty
*Department of Computer Science*
*Florida State University*

***Abstract*—In recent years, deep learning has revolutionized the field of computer vision and has achieved state-of-the-art performance in a variety of applications. However, training a robust deep neural network necessitates a large amount of hand-labeled training data, which is time-consuming and labor-intensive to acquire. Active learning and transfer learning are two popular methodologies to address the problem of learning with limited labeled data. Active learning attempts to select the salient and exemplar instances from large amounts of unlabeled data; transfer learning leverages knowledge from a labeled source domain to develop a model for a (related) target domain, where labeled data is scarce. In this paper, we propose a novel active transfer learning algorithm with the objective of learning informative feature representations from a given dataset using a deep convolutional neural network, under the constraint of weak supervision. We formulate a loss function relevant to the research task and exploit the gradient descent algorithm to optimize the loss and train the deep network. To the best of our knowledge, this is the first research effort to propose a task-specific loss function integrating active and transfer learning, with the goal of learning informative feature representations using a deep neural network, under weak human supervision. Our extensive empirical studies on a variety of challenging, real-world applications depict the merit of our framework over competing baselines.**

***Index Terms*—active learning, transfer learning, deep learning, image recognition**

## I. Introduction

Deep learning algorithms automatically learn a discriminating set of features and have depicted commendable performance in a variety of applications. Architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Generative Adversarial Networks (GANs) etc., have created a paradigm shift in computer vision applications and have depicted tremendous performance improvements in several tasks, including image recognition [1], object detection [2], multimodal emotion recognition [3] and image segmentation [4] among others. However, training a deep neural network requires a large volume of labeled training data, acquiring which is an expensive process in terms of time, labor and human expertise. Thus, developing intelligent machine learning models under the constraint of weak human supervision has attracted significant research attention in recent years.

*Active learning* (AL) and *transfer learning* (TL) or *domain adaptation* (DA) are two popular techniques to address the problem of limited labeled data. AL algorithms automatically identify the most informative samples from a large collection of unlabeled data. This tremendously reduces human annotation effort, as only a few samples that are identified by the algorithm, need to be labeled manually. Further, since the model gets trained on the exemplar instances, its generalization capability is typically much better than a standard passive learner, where the training data is sampled at random from the underlying population [5]. TL algorithms handle the problem of learning with weak supervision by utilizing abundant labeled data in one domain to develop a model for a related domain of interest, where there is a paucity of labeled data [6]. The domain of interest is referred to as the *target* domain and the other domain is called the *source* domain. The probability distributions generating the data in the two domains are different, which implies a difference in their joint probability distributions: $P_S(X,Y) \neq P_T(X,Y)$. Since target domain samples are scarce, it is challenging to accurately compute $\widehat{P_T}(X,Y)$. The main objective of DA is to approximate the distribution $\widehat{P_T}(X,Y)$ using information from the source domain, in order to develop an accurate prediction model for the target domain. To this end, the source and target domains are assumed to be correlated, where $P_S(X) \neq P_T(X)$ but $P_S(Y|X) \approx P_T(Y|X)$ ; that is, the marginal distributions of the source and target are different, but their conditional distributions are the same [6]. Both active learning and transfer learning have been used with remarkable success in a variety of computer vision applications [7]–[10].

Even though there have been a few research efforts to combine AL and TL, none of them have specifically focused on the problem of training a deep learning model, with the goal of learning informative feature representations from a given dataset. In this paper, we propose a novel framework called *Deep Active Transfer Learning* (DATL), to address this challenge. Specifically, we attempt to answer the following research question: *We attempt to train a deep CNN in a target domain of interest. We are given $N_T^L$ labeled samples and $N_T^U$ unlabeled samples in the target domain ($N_T^L \ll N_T^U$). We are also given $N_S$ labeled samples in a related source domain; however, there is a probability distribution difference between the source and the target. A query budget $k$ is given, which denotes the number of labels that can be purchased in the target domain. Which $k$ samples should we select from the set of unlabeled target samples, in order to induce a deep CNN with maximum generalization capability?* This is depicted in

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

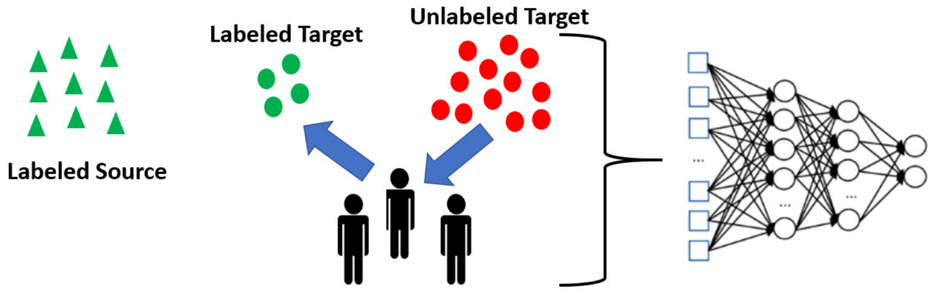


Fig. 1. Active Transfer Learning problem setup. We are given labeled data (green triangles) from a source domain, a small amount of labeled data (green circles) and a large amount of unlabeled data (red circles) from a target domain. There is a probability distribution difference between the two domains (denoted by the triangles and the circles). We are allowed to query the labels of $k$ unlabeled target samples. Our objective is to select the $k$ most informative samples so that a deep CNN trained on this data has maximum generalization capability. Best viewed in color.

Figure 1. Specifically, we attempt to address the following three challenges through a single integrated framework:

- Leverage the labeled data in the source domain by addressing the probability distribution difference between the source and the target
- Identify the most exemplar unlabeled target samples for manual annotation
- Learn informative feature representations from the data using a deep neural network (CNN)

To this end, we propose a novel loss function encompassing the aforementioned challenges and train the CNN by optimizing the loss using gradient descent. This is the first research effort to integrate AL and TL through a joint, task-specific loss function, with the goal of reducing the human annotation effort in inducing a deep learning model. Although validated on vision data in this research, the proposed framework is generic and can be used in any application to optimize the human effort in training deep models. The rest of the paper is organized as follows: we present a survey of related techniques in Section II; the details of our framework are presented in Section III; our empirical studies are detailed in Section IV; and we conclude with discussions in Section V.

## II. Related Work

In this section, we present a survey of active learning, transfer learning and the few active transfer learning algorithms developed to combine the two methods.

**Active Learning:** Active learning is a well-researched topic in the machine learning literature. Pool-based batch mode active learning (BMAL) is the most common variant where the learner is exposed to a pool of unlabeled samples and it iteratively queries samples for annotation. Uncertainty sampling is the most common active learning strategy, where unlabeled samples furnishing the maximal classification uncertainty are queried for annotation. The uncertainty of an unlabeled sample can be computed in many different ways, such as Shannon's entropy [11], its distance from the decision boundary for SVM models [12], the extent of disagreement among a committee of classifiers regarding its label [13], and the expected model change [14] among others. Multiple criteria such as uncertainty, representativeness and diversity can also be combined to quantify the information content of an unlabeled sample [15]. The Fisher information matrix has also been exploited as a metric of model uncertainty to develop AL algorithms [10]. Matrix partitioning techniques have been studied to identify a batch of informative unlabeled samples for AL [16]. Adversarial techniques using GANs have also been used for active learning [17]. Recently, there has been a body of research focusing on novel extensions of AL such as actively completing a data matrix [18], active video summarization [19], active learning with novel query types [20] and active learning with imperfect / noisy oracles [21] among others.

**Transfer Learning:** Domain adaptation (DA) or transfer learning is also a well-studied problem in machine learning. Before deep learning became popular, researchers primarily relied on hand-crafted features for DA [22], [23]. DA techniques using deep models have outperformed their non-deep counterparts due to the highly informative feature representations learned by the deep networks. Tzeng *et al.* [7] proposed the Deep Domain Confusion (DDC) algorithm where the Maximum Mean Discrepancy (MMD) was used to quantify the domain disparity and learn domain invariant features. Long *et al.* [24] proposed the Deep Adaptation Networks (DAN) model where the MMD loss was applied in all the fully connected layers (*fc6, fc7 and fc8*) of the AlexNet, with promising empirical performance. The Residual Transfer Network (RTN) architecture, proposed by Long *et al.* [25], incorporated a residual layer in the network and used MMD to address domain disparity. DA has also been used to learn discriminating hash codes for the source and target data, while addressing the probability distribution difference between them [26]. The Generative Adversarial Networks (GANs) proposed by Goodfellow *et al.* [27] is one of the hallmarks of deep learning research. Several recent techniques have explored adversarial training for domain adaptation, such as the Domain Adversarial Neural Network (DANN) which incorporates a domain classifier, whose gradient is reversed when learning the feature extractor weights [28], the Coupled Generative Adversarial Network (CoGAN) model, which shares weights at different layers of the GAN to train a coupled network, and the combination of CoGAN with Variational Autoencoder (VAE) [29] to develop an image translation network [8] among others. Concepts from Wasserstein GAN have also been used for domain adaptation [30]. The adversarial methods based on GANs have depicted commendable empirical performance.

**Active Transfer Learning:** Even though AL and TL are extensively studied separately, there have been relatively few research efforts to combine the two methodologies. Initial research efforts in this direction performed transfer and active learning in two separate stages, which may cause redundancy or information overlap between the instances selected from the source and target domain data [31]–[33]. For instance, Saha *et al.* [34] combined uncertainty region sampling with several transfer learning concepts and provided an analysis of label complexity and error rates. A Bayesian framework for active transfer learning was proposed in [35], based on prior-dependent learning. Chattopadhyay *et al.* [36] proposed JOTAL – an integrated framework that performs transfer and active learning simultaneously by solving a single convex optimization problem. The framework computes the weights of source domain data and selects the samples from the target domain data simultaneously, by minimizing a common objective of reducing distribution difference between the data set consisting of re-weighted source and the queried target domain data, and the set of unlabeled target domain data. Kale and Liu proposed the Transfer-accelerated, Importance Weighted Consistent Active Learning (TIWCAL) algorithm, where the main idea was to use transfer learning to initialize the active learner using data from a related task [37]. The active learner can thus make more informed queries in the early rounds, which can potentially address the cold-start problem. The same authors also proposed the Hierarchical Active Transfer Learning (HATL) algorithm which exploits the cluster structure of the data shared between the source and target domains to perform transfer learning by imputing the labels of the unlabeled target data, and to generate effective label queries during active learning [38]. Active transfer learning has also been used in regression applications, such as recommender systems [39] and estimating the yield of vineyards from images of grapes [40]. Even though the fields of active learning and transfer learning have significantly progressed independently (especially with the growing popularity of deep learning), active transfer learning has not progressed much beyond the aforementioned research.

All the active transfer learning methods work on hand-engineered features which need to be supplied as an input to the algorithms. Motivated by the unparalleled success of deep neural networks to learn informative feature sets, we propose an active transfer learning framework, specifically tailored to train deep models. Our framework can address the domain disparity between the source and the target, identify the exemplar samples from the unlabeled target data for manual annotation, and simultaneously learn a representative set of features from the data using a deep CNN. We now describe our framework.

## III. Proposed Framework

As shown in Figure 1, we are given data from two domains: source and target. The data in the source domain are all labeled: $D_S = \{x_i, y_i\}_{i=1}^{N_S}$. In the target domain, we are given a small number of labeled samples: $D_T^L = \{x_j, y_j\}_{j=1}^{N_T^L}$ and a large number of unlabeled samples: $D_T^U = \{x_j\}_{j=1}^{N_T^U}$. We are also given a budget $k$, which denotes the number of samples that can be labeled from the unlabeled target set. Our objective is to select the $k$ most informative samples from $D_T^U$ and get them labeled by human annotators, so that a deep neural network trained on this data has maximum generalization capability on unseen test data from the target domain. Instead of using an off-the-shelf network trained on a different dataset, for a different application, we propose to formulate a novel loss function specific to the application in question and train the network to optimize that loss. Our network will then get specifically tailored to our application and can potentially depict improved learning performance. Our loss function consists of three components: $(i)$ supervised loss on labeled data, which encourages the network to be consistent with the labeled data, that is, incur minimal prediction error on the labeled source and target samples; $(ii)$ a strategy to address the disparity between the source and target domains and learn feature representations accordingly; and $(iii)$ unsupervised loss on unlabeled target data, which encourages the network to deliver high confidence predictions on the unlabeled target set. These are detailed below:

### A. Supervised Loss on Labeled Data

The goal of this term is to ensure that the network furnishes accurate predictions on the labeled data. Let $D^L = D_S \cup D_T^L = \{x_1, x_2, \dots, x_{n_L}\}$ be the labeled source and target data with corresponding labels $\{y_1, y_2, \dots, y_{n_L}\}$. We used the standard cross-entropy (CE) loss to estimate the classification error:

$$L_{CE} = \frac{1}{n_L} \sum_{i=1}^{n_L} L(f(x_i), y_i)$$

where

$$L(f(x_i), y_i) = -\sum_{j=1}^{C} \mathbf{1}(y_i = j) \log f_j(x_i) \tag{1}$$

Here $C$ is the total number of classes, $\mathbf{1}$ is the indicator function and $f(x_i) = [f_1(x_i), f_2(x_i), \dots, f_C(x_i)]^T$ is a probability vector (obtained using the softmax activation of the deep network) with $f_j(x_i)$ being the probability that sample $x_i$ is assigned to category $j$.

### B. Disparity between Source and Target Domains

Our strategy to address the domain disparity between the source and the target is inspired by the adversarial learning framework proposed by Ganin *et al.* [28]. Our objective is to learn features in such a way that a domain classifier trained to differentiate source and target samples has high error, that is, source and target domain features become indistinguishable.

This is implemented by incorporating a cross entropy loss $L_d$ on the domain classifier and maximizing it (since we desire high error) using stochastic gradient descent (SGD). Ganin *et al.* introduced the concept of a gradient reversal layer (GRL) between the feature extractor and the domain classifier. The GRL acts as an identity transformation during forward propagation; during back-propagation, it takes the gradient from the subsequent level and multiplies it by $-1$ before passing it to the preceding layer. Running SGD in this model essentially becomes equivalent to updating the weight parameters of the deep network. Please refer [28] for more details about this algorithm.

### *C. Unsupervised Loss on Unlabeled Target Data*

We propose a class alignment (CA) loss on the unlabeled target data to ensure that the network furnishes high confidence predictions on the unlabeled target set. Since this is a multi-class problem, each unlabeled target sample can belong to exactly one of the $C$ classes. We assume the presence of $M$ samples from each class $j$ in the labeled source data, where $j \in \{1, 2, \ldots, C\}$, and let $w_S^{jm}$ be the $m^{th}$ source output from class $j$. The fundamental idea is to ensure that the output $w_T^i$ of an unlabeled target sample $x_i$ is similar to all the $M$ source outputs from a particular class $j$ and dissimilar to all the other classes (we used the dot product to compute similarity). Enforcing similarity with all the $M$ data points results in a more robust target data class assignment. We define a measure to capture this idea, which quantifies the probability that the target sample $x_i$ is assigned to class $j$:

$$p_{ij} = \frac{\sum_{m=1}^{M} exp\langle w_T^i, w_S^{jm}\rangle}{\sum_{c=1}^{C}\sum_{m=1}^{M} exp\langle w_T^i, w_S^{cm}\rangle} \tag{2}$$

Here, $\langle \cdot, \cdot \rangle$ denotes the dot product between two vectors, the exponential function $exp(.)$ has been used for ease of differentiability and the denominator ensures that the meaure is normalized, that is, $\sum_j p_{ij} = 1$. When the output of the target sample is similar to exactly one class and dissimilar to all the other classes, the probability vector $p_i = [p_{i1}, p_{i2}, \ldots, p_{iC}]$ tends to be a one-hot vector, with one entry high and the others low. This implies that the unlabeled target sample aligns well with exactly one of the classes, and can thus be interpreted as having low prediction uncertainty (entropy). The class alignment loss is therefore defined to capture the entropy of the target probability vectors:

$$L_{CA} = -\frac{1}{n_T^U}\sum_{i=1}^{n_T^U}\sum_{j=1}^{C} p_{ij}\log p_{ij} \tag{3}$$

where $n_T^U$ denotes the number of unlabeled target samples. Minimizing this loss produces probability vectors $p_i$ that tend to be one-hot vectors, that is, the unlabeled target data sample outputs are similar to source data outputs from one and only one class. This ensures that the deep network furnishes confident predictions on the unlabeled target data. Computing the similarity with $M$ source samples ensures that the feature representations are learned based on a common similarity between multiple source category data points and the target data point. Note that the probability values in Equation (3) are derived using the novel class alignment score in Equation (2) and not using class prediction probabilities, as done conventionally.

The overall loss function to train the deep network can thus be written as:

$$L = L_{CE} - \lambda_1 L_d + \lambda_2 L_{CA} \tag{4}$$

where $\lambda_1$ and $\lambda_2$ are weights governing the relative importance of the terms. Since the overall loss function needs to be minimized, we have a negative sign in front of $L_d$, as we would like to maximize the cross entropy loss of the domain classifier (as explained in Section III-B). SGD was used minimize the loss and train the network.

### *D. Query Strategy for Active Learning*

Once the deep CNN was trained, the unlabeled target samples were passed through the network and the value of the class alignment loss $L_{CA}$ (Equation (3)) was computed for each sample. Since the network was trained to minimize the class alignment loss, the unlabeled target samples furnishing the highest values of this term were deemed the most informative from an active learning perspective. The top $k$ samples sorted by the $L_{CA}$ score were therefore queried for manual annotation.

### *E. Network Architecture*

The architecture of the CNN used in this study is shown in Figure 2. It consists of 3 components, a feature extractor, a domain classifier and a label classifier. We used two convolutional layers with filters of size $3 \times 3$ as the feature extractor. The input images to the first convolutional layer were scaled to $128 \times 128$ pixels. Each of the convolutional layer is followed by *tanh* activation function and a max-pooling layer performing spatial pooling over a $3 \times 3$ window. A batch normalization layer follows to reduce the shift in the hidden unit values. The domain classifier is composed of a gradient reversal layer (as discussed in Section III-B) followed by a fully connected layer with 128 units and a *tanh* activation function. A batch normalization layer and dropout regularization layer follow the fully connected layer. The *sigmoid* output layer predicts the domain of the sample. The input to the category / label classifier comes from the feature extractor output. This component of the network comprises of a fully connected layer with 128 units followed by a *tanh* activation function and an output *softmax* layer which predicts the label of the sample. The network was trained for $50$ epochs using the *SGD* optimizer with a learning rate of $10^{-5}$. We used a Google cloud compute instance with a n1-standard-4 4vCPUs machine type with 15 GB memory. The instance had a NVIDIA Tesla P100 GPU with 16 GB memory attached to it, accelerating the processing and training of our model.

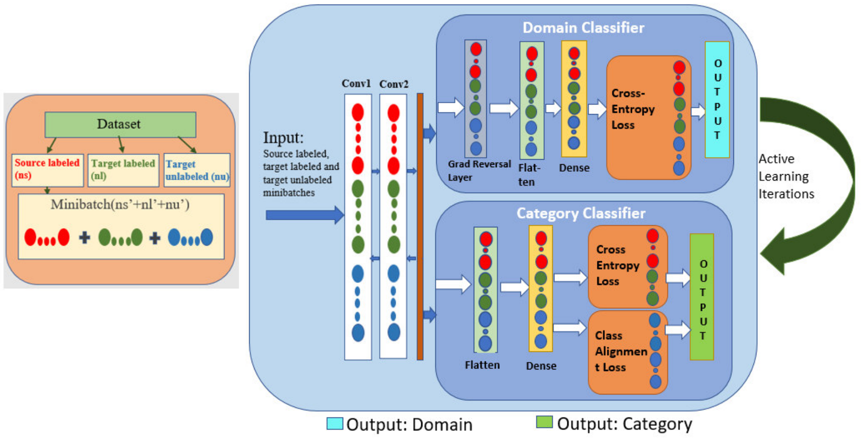


Fig. 2. Architecture of the deep convolutional neural network used in our study. The network is trained using mini-batches consisting of source samples (red), labeled target samples (green) and unlabeled target samples (blue). The cross-entropy loss for the domain classifier operates on all three types of samples; the cross-entropy loss for the category / label classifier operates only on the labeled source and target samples; the class alignment loss operates only on the unlabeled target samples. Best viewed in color.

## IV. Experiments and Results

In this section, we present an empirical analysis of our framework against relevant baselines. Besides the learning peformance, we studied the effects of the size of the labeled source set, labeled target set and batch size on the predictive performance. These are presented in the following sections.

### A. Datasets and Feature Extraction

We validated the performance of DATL on 9 transfer tasks (source-target pairs) across 4 datasets. These are detailed below. Note that our objective in this research was to study the performance of active transfer learning algorithms, and not to outperform the state-of-the-art error rates on these datasets. We therefore did not replicate the exact train/test splits used in previous research on these datasets, where the objective was to achieve the lowest prediction error rates.

**Office:** This is a popular benchmark dataset for object recognition in the DA computer vision community [41]. It contains images of everyday objects in an office environment and has 3 domains: Amazon, Webcam and DSLR.

**Office-Home:** This dataset was recently introduced and has 4 domains: Art (artistic depictions of objects in the form of sketches, paintings, ornamentation, etc.), Clipart (collection of clipart images), Product (images of objects without a background, akin to the Amazon category in the *Office* dataset) and Real-World (images of objects captured with a regular camera) [26].

**MNIST and USPS:** We also studied the performance of our framework on two handwritten digits datasets, MNIST and USPS [42], which contain images of handwritten digits from 10 classes. Both these datasets are extensively used in computer vision research. While in the previous two datasets we study the performance on different domains within a dataset, here we study the performance across two different datasets, to comprehensively evaluate our proposed method.

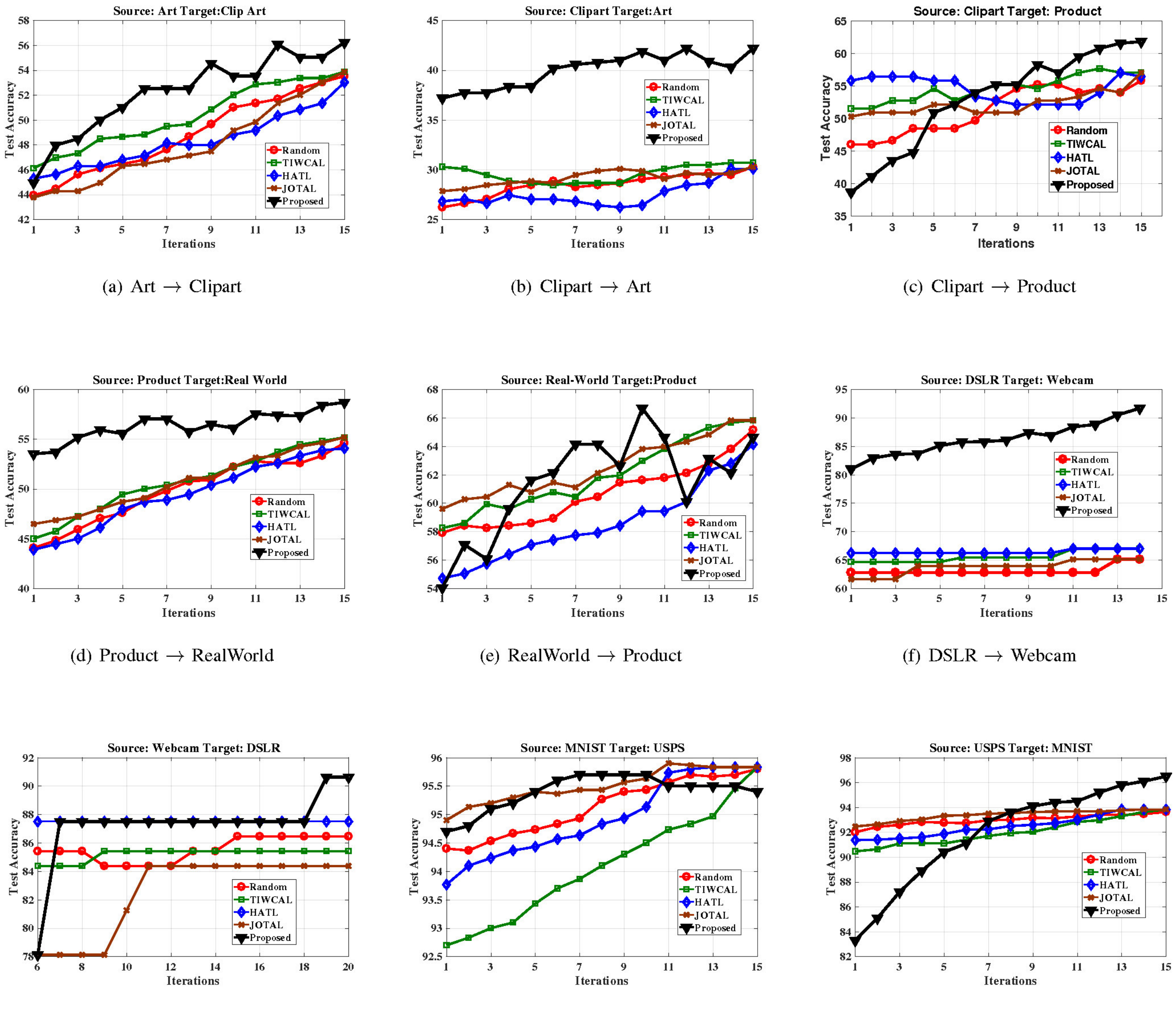
**Feature Extraction:** The comparison baselines used in this research (detailed below) work only on hand-crafted features. For fair comparison, we extracted deep features (using our untrained network) from each image and passed them as inputs to the baseline methods.

### B. Experimental Setup

In each experiment, we are given a source set and a target set. The target set was divided into three parts: an initial labeled training set, an unlabeled set and a test set. The number of labeled target samples was much less than the number of unlabeled target samples, to appropriately mimic a real-world application. For a given batch size $k$, each algorithm queried $k$ samples from the unlabeled target set, which were labeled and appended to the labeled set. The underlying classification model was updated and the accuracy was evaluated on the test set. The process was continued until some stopping criterion was satisfied (taken as 15 iterations in this work). Similar to all the previous active transfer learning research, our objective was to study the growth in accuracy on the target test set, with increasing number of iterations. Each experiment was conducted 3 times (with different initial training, unlabeled and test sets) and the results were averaged to rule out the effects of randomness. The parameters $\lambda_1$ and $\lambda_2$ were both taken as 1 based on preliminary experiments.

### C. Comparison Baselines

The problem setup of active transfer learning is depicted in Figure 1 and is different from both active and transfer learning. To facilitate fair comparison, we used the following active transfer learning algorithms as comparison baselines in our work (and not algorithms that are designed for only active learning or only transfer learning): $(i)$ **Random Sampling**: a batch of $k$ samples is selected at random from the unlabeled



(a) Art → Clipart

(b) Clipart → Art

(c) Clipart → Product

(d) Product → RealWorld

(e) RealWorld → Product

(f) DSLR → Webcam

(g) Webcam → DSLR

(h) MNIST → USPS

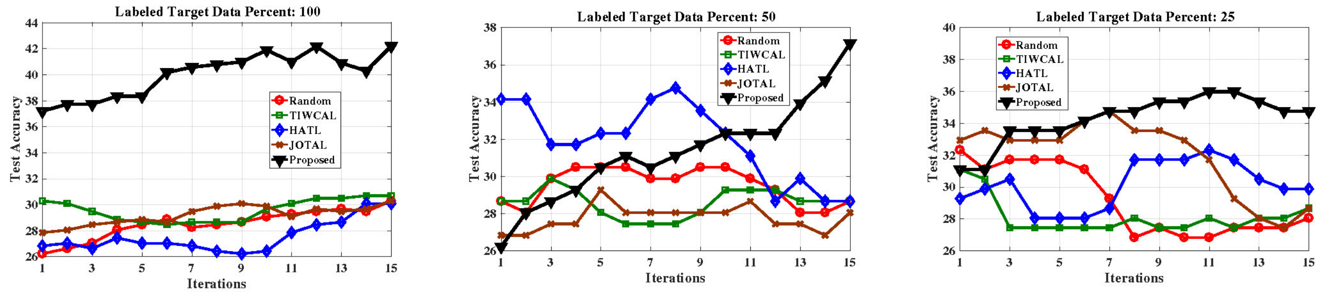
(i) USPS → MNIST

Fig. 3. Active transfer learning performance comparison on 9 transfer tasks. The $x$-axis denotes the number of iterations and the $y$-axis denotes the classification accuracy. The notation $A \rightarrow B$ implies $A$ is the source and $B$ is the target. Best viewed in color.

target set; $(ii)$ **TIWCAL**: the Transfer-accelerated, Importance Weighted Consistent Active Learning proposed by Kale and Liu [37]; $(iii)$ **HATL**: the Hierarchical Active Transfer Learning algorithm proposed by Kale *et al.* [38]; and $(iv)$ **JOTAL**: the joint transfer active learning framework proposed by Chattopadhyay *et al.* [36]. These baselines were selected as they are the top performing active transfer learning algorithms for classification problems [38].

### *D. Learning Performance*

The performance results are depicted in Figure 3. In each graph, the $x$-axis denotes the number of iterations and the $y$-axis denotes the accuracy on the test set (the notation $A \rightarrow B$ implies $A$ is the source and $B$ is the target). We note that the performance of the baseline methods is inconsistent across the experiments. For instance, *JOTAL* depicts impressive performance for the MNIST → USPS experiment, but much inferior performance for the Art → Clipart experiment. Similarly, the *HATL* and *TIWCAL* methods perform well in the Clipart → Product experiment, but the performance drastically drops for the DSLR → Webcam experiment. *Random Sampling* also exhibits inconsistent accuracy growth. The proposed *DATL* method consistently shows impressive performance across all the experiments. The improvement in performance over the baselines is particularly evident for the Art → Clipart, Clipart → Art, Product → RealWorld and DSLR → Webcam experiments. It also furnishes the highest accuracy after 15

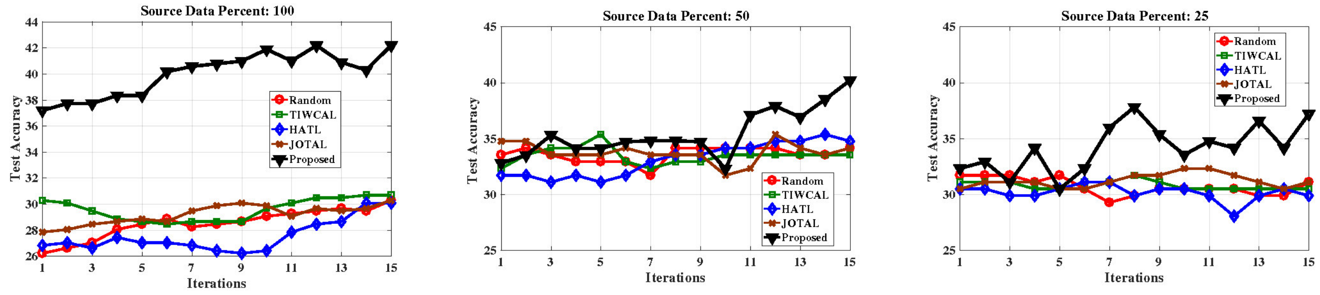


(a) Labeled Target Data Percent: 100

(b) Labeled Target Data Percent: 50

(c) Labeled Target Data Percent: 25

Fig. 4. Effect of Labeled Target Data for the Clipart → Art experiment. The result with $100\%$ labeled target data is the same as that in Figure 3(b) and is included for reference. Best viewed in color.



(a) Source Data Percent: 100

(b) Source Data Percent: 50

(c) Source Data Percent: 25

Fig. 5. Effect of Source Data for the Clipart → Art experiment. The result with $100\%$ source data is the same as that in Figure 3(b) and is included for reference. Best viewed in color.

iterations in 7 out of the 9 experiments. Since our *DATL* framework learns features by optimizing a loss specific to the problem, it depicts much better performance than other active transfer learning algorithms, where the features are not learned from the data. The results unanimously support the fact that *DATL* is instrumental in reducing human annotation effort to induce a robust deep learning model, and corroborates its usefulness in real-world applications.

### *E. Effect of Labeled Target Data*

The goal of this experiment was to study the effect of labeled target data on the learning performance. The results for the experiment Clipart → Art are depicted in Figure 4, with varying percentages of labeled target data ($50\%$ and $25\%$). *DATL* depicts impressive performance compared to all the baselines, across all experiments and achieves a much higher accuracy after 15 iterations. The results in Figure 4(c) with $25\%$ labeled target data is particularly relevant, as there is typically a scarcity of labeled data in the target domain in most real-world applications. This shows the robustness of our framework to the amount of labeled target data.

### *F. Effect of Source Data*

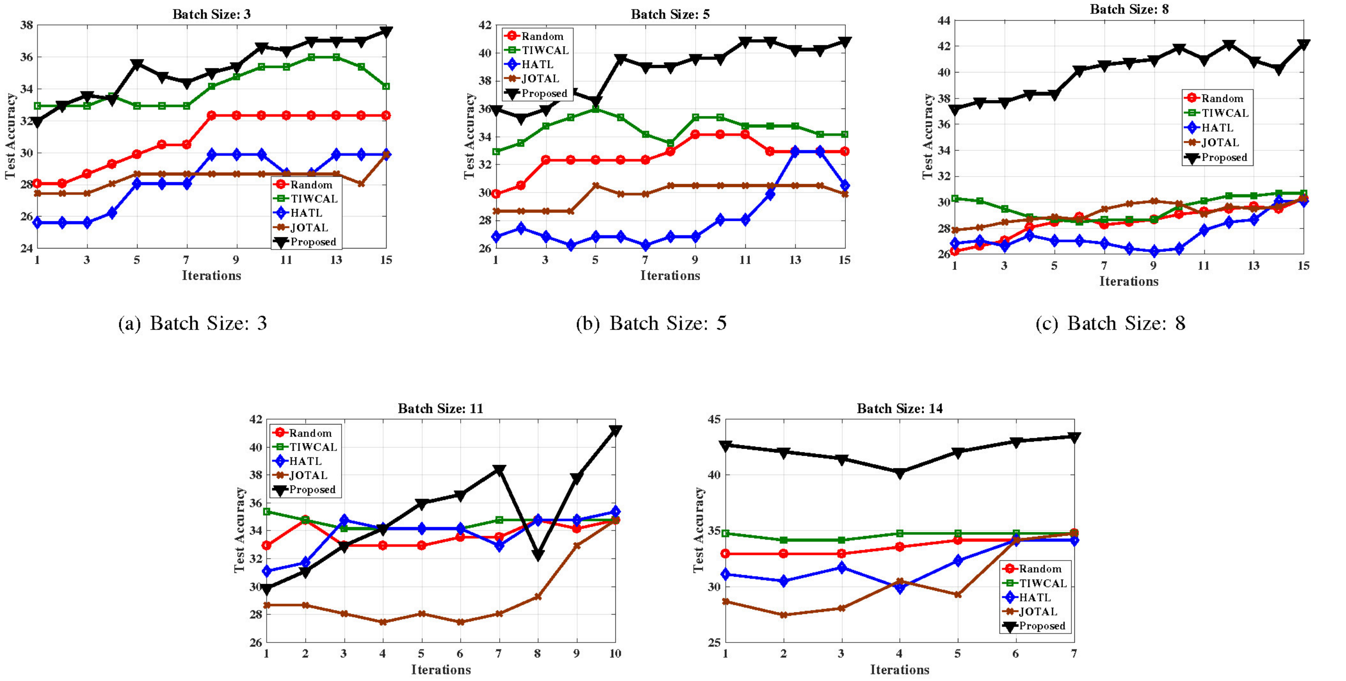
In this experiment, we studied the effect of source data on the learning performance. Figure 5 presents the results on the Clipart → Art experiment with varying percentages of source data ($50\%$ and $25\%$). *DATL* once again outperforms the baselines, particularly in the later iterations. This corroborates the flexibility of our framework to varying sizes of the source dataset.

### *G. Effect of Batch Size*

The effect of the batch size $k$ on the learning performance was studied in this experiment. The results are presented in Figure 6, for batch sizes $3, 5, 8, 11$ and $14$. *DATL* consistently depicts superior performance across all batch sizes. This once again shows the usefulness of our framework for real-world applications where the batch size is governed by the available labeling resources, and is different for different applications.

## V. Conclusion and Future Work

In this paper, we proposed a novel deep active transfer learning (DATL) framework to reduce human annotation effort in training a deep CNN. This is an extremely relevant problem in today's digital world where there is an ever increasing demand of labeled data to train deep models, but a serious dearth of human labor to hand-label data samples. We proposed a novel loss function specifically tailored to the research task and exploited the gradient descent algorithm to optimize the loss and train the deep network. Our extensive empirical studies



(a) Batch Size: 3

(b) Batch Size: 5

(c) Batch Size: 8

(d) Batch Size:11

(e) Batch Size: 14

Fig. 6. Effect of Batch Size for the Clipart → Art experiment. The result with Batch Size = 8 is the same as that in Figure 3(b) and is included for reference. Best viewed in color.

on 9 tasks corroborated the merit of DATL over competing baselines, in terms of the learning performance, as well as the effects of the number of labeled source and target samples and the batch size. As part of future research, we plan to study the performance of our framework on other popular deep architectures such as LSTMs, RNNs, GANs etc.

## References

[1] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Neural Information Processing Systems (NIPS)*, 2012.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[3] H. Ranganathan, S. Chakraborty, and S. Panchanathan, “Multimodal emotion recognition using deep learning architectures,” in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[5] B. Settles, “Active learning literature survey,” in *Technical Report: University of Wisconsin-Madison*, 2010.

[6] S. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 22, no. 10, 2010.

[7] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[8] M. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” in *Advances of Neural Information Processing Systems (NIPS)*, 2017.

[9] S. Chakraborty, V. Balasubramanian, Q. Sun, S. Panchanathan, and J. Ye, “Active batch selection via convex relaxations with guaranteed solution bounds,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 37, no. 10, pp. 1945–1958, 2015.

[10] S. Hoi, R. Jin, J. Zhu, and M. Lyu, “Batch mode active learning and its application to medical image classification,” in *International Conference on Machine Learning (ICML)*, 2006.

[11] A. Holub, P. Perona, and M. Burl, “Entropy-based active learning for object recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR-W)*, 2008.

[12] S. Tong and E. Chang, “Support vector machine active learning for image retrieval,” in *Proceedings of the ninth ACM international conference on Multimedia*, 2001.

[13] Y. Freund, S. Seung, E. Shamir, and N. Tishby, “Selective sampling using the query by committee algorithm,” *Machine Learning*, vol. 28, no. 2-3, pp. 133–168, 1997.

[14] W. Cai, Y. Zhang, and J. Zhou, “Maximizing expected model change for active learning in regression,” in *IEEE International Conference on Data Mining (ICDM)*, 2013.

[15] D. Shen, J. Zhang, J. Su, G. Zhou, and C. Tan, “Multi-criteria based active learning for named entity recognition,” in *Association for Computational Linguistics (ACL)*, 2004.

[16] Y. Guo, “Active instance sampling via matrix partition,” in *Advances of Neural Information Processing Systems (NIPS)*, 2010.

[17] J. Zhu and J. Bento, “Generative adversarial active learning,” in *Advances of Neural Information Processing Systems (NIPS) Workshops*, 2017.

[18] N. Ruchansky, M. Crovella, and E. Terzi, “Matrix completion with queries,” in *ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2015.

[19] A. Molino, X. Boix, J. Lim, and A. Tan, “Active video summarization: Customized summaries via on-line interaction with the user,” in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2017.

[20] S. Xiong, Y. Pei, R. Rosales, and X. Fern, “Active learning from relative comparisons,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 12, 2015.

[21] S. Yan and K. Chaudhuri, “Active learning from imperfect labelers,” in *Advances of Neural Information Processing Systems (NIPS)*, 2016.

[22] S. Pan, I. Tsang, J. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2009.

[23] D. Pardoe and P. Stone, "Boosting for regression transfer," in *International Conference on Machine Learning (ICML)*, 2010.

[24] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International Conference on Machine Learning (ICML)*, 2015.

[25] M. Long, H. Zhu, J. Wang, and M. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Advances of Neural Information Processing Systems (NIPS)*, 2016.

[26] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[27] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances of Neural Information Processing Systems (NIPS)*, 2014.

[28] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research (JMLR)*, vol. 17, 2016.

[29] D. Kingma and M. Welling, "Auto-encoding variational bayes," in *arXiv preprint arXiv:1312.6114*, 2013.

[30] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

[31] Y. Chan and H. Ng, "Domain adaptation with active learning for word sense disambiguation," in *Association of Computational Linguistics (ACL)*, 2007.

[32] X. Shi, W. Fan, and J. Ren, "Actively transfer domain knowledge," in *European Conference on Machine Learning (ECML)*, 2008.

[33] P. Rai, A. Saha, H. Daume, and S. Venkatasubramanian, "Domain adaptation meets active learning," in *NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, 2010.

[34] A. Saha, P. Rai, H. Daume, S. Venkatasubramanian, and S. DuVall, "Active supervised domain adaptation," in *European Conference on Machine Learning (ECML)*, 2011.

[35] L. Yang, S. Hanneke, and J. Carbonell, "A theory of transfer learning with applications to active learning," *Machine Learning*, vol. 90, no. 2, 2012.

[36] R. Chattopadhyay, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, "Joint transfer and batch mode active learning," in *International Conference on Machine Learning (ICML)*, 2013.

[37] D. Kale and Y. Liu, "Accelerating active learning with transfer learning," in *IEEE International Conference on Data Mining (ICDM)*, 2013.

[38] D. Kale, M. Ghazvininejad, A. Ramakrishna, J. He, and Y. Liu, "Hierarchical active transfer learning," in *SIAM Data Mining Conference (SDM)*, 2015.

[39] L. Zhao, S. Pan, E. Xiang, E. Zhong, Z. Lu, and Q. Yang, "Active transfer learning for cross-system recommendation," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2013.

[40] X. Wang, T. Huang, and J. Schneider, "Active transfer learning under model shift," in *International Conference on Machine Learning (ICML)*, 2014.

[41] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European Conference on Computer Vision (ECCV)*, 2010.

[42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of IEEE*, 1998.