Multi-Source Learning based on a feature transferable framework

Jingge Wang[1], Jihong Wang[1] and Jiarong Li*[1]
1. DSIT Research Center, Tsinghua-Berkeley Shenzhen Institute, Shenzhen, 518055, PR China
(*Email: jiarongli.00@gmail.com)

***Abstract*—The machine learning and computer vision community is witnessing an unprecedented rate of new tasks being proposed and addressed in , thanks to the power of deep convolutional neural network to find complex mappings from feature space X to label space Y [1]. The advent of each task often accompanies the release of a large-scale human-labeled dataset, for supervised training of the deep network. However, it is expensive and time-consuming to manually label sufficient amount of training data. Therefore, to gain useful knowledge for the target mission, it is important to develop algorithms that can exploit off-the-shelf labeled data set to get knowledge about target domain. While previous works focus primarily on transferring learning from a single source, we are researching multi-source transferring across domains and tasks. In this paper,we propose a new framework which achieves multi-sources transfer-learning by training a classifier to re-weight various sources in order to adapt the rich yet complex information among sources to boost the target learning. Experiments of two source domains transferring to one target domain illustrate the effectiveness of our method.**

***Index Terms*—multi-source transfer learning; deep learning; convolutional neural network; classification**

I. Introduction

Machine learning has advanced dramatically over the past two decades and has become a practical technology for widespread commercial use from the laboratory. Machine learning is currently one of the fastest growing technologies at the heart of artificial intelligence and data science, commonly used in intrusion detection, speech recognition, computer vision, pattern recognition, text analysis, and other fields. It's got great results, of course. However, many machine learning algorithms need to satisfy the following two basic conditions in order to obtain a high accuracy classification model: (1) the training and test data come from the same feature space with the same distribution, which fulfill the independent and identical distribution conditions; (2) There are enough examples [2] available for training. Nonetheless, in practical applications, these expectations are not always easily met. Particularly in emerging applications such as text mining, bioinformatics, distributed network sensor networks, and social network science, where the independent and identical distribution conditions between training and testing data can not be fulfilled under the influence of time, environmental changes, or sensor system instability. When the distribution of data shifts, we need to re-collect the training data in most models, while the previous training data won't be used again, resulting in unused data resources. Furthermore, data sample resources are often scarce in some areas, and data collection costs are very expensive or even impossible. In this case, it is desirable to transfer knowledge between fields of work.

Transfer learning, also known as domain adaptation, is an important way of solving the above-mentioned problems. On the one hand, in order to satisfy independent and equivalent distribution conditions, we no longer requires new training and testing data. On the other hand, when training data in the target domain is sparse and not adequate to obtain a good classifier, while the source domain data (often containing a large number of labeling samples) is similar to the target domain, source domain can be used to assist the learning tasks in the target domain.

II. Related work

Although domain adaptation has been extensively studied in recent years, most theoretical studies and algorithms focus on the setting of single-source single-target adaptation, which results that the effectiveness of target domain predictors is largely dependent on transferability between source domain and target domain. If there is a strong transferability between the two domains, there is no question that the transfer learning approach can do a good job in extracting such relevant information and providing a predictive model in the training process of the target domain. Nevertheless, if there is no similarity between the source domain and the target domain or the similarity is low, then the transfer learning approach can not improve the performance of target domain predictors and may even result in a negative transfer that reduces the performance of the target domain predictor.

Previous multi-source transfer learning [3]–[5] focuses on extracting domain-free representations from multiple sources rather than combining them together.There are typically two approaches to tackle the learning of multi-source conversion. (1) One strategy is to re-weigh different sources to adapt the rich yet complex information between sources to enhance target learning [3], [4]. (2) Another successful strategy is to use the multi-task framework to jointly guide knowledge transfer with multiple sources [5], [6].

In this project, we implement multi-source transfer learning by re-weighting different sources to adapt the rich but complex information between sources to improve the performance of target learning. We experiment by using large and relative small amount of source data to transfer to a target domain respectively, and results illustrate the effectiveness of our proposed multi-source method.

The following is the structure of this article. Section 2 outlines the related work other scholar did. In Section 3, first we introduce the data set and model we used, and then we explain the structure of our model. Section 4 outlines the related experiments we did and explores the insight of our model. Section 5 draws a conclusion.

III. A FEATURE TRANSFERABLE FRAMEWORK FOR MULTI-SOURCE LEARNING

In this section, we proposed a feature transferable framework for multi-source learning by re-weighting different sources. First of all, we'll start with some basic knowledge.

A. Dataset

It is well-known that deep models require a large number of training data. However, existing data set for visual tasks are usually small-scale or limited in the number of categories, so it's very important to transfer knowledge from relevant sources to help us train a model. By far, we have Food-256 [7], which contains 256 kinds of food photos. Each food photo has a bounding box indicating the location of the food item in the photo. Most of the food categories in this dataset are popular foods in Japan and other countries. We also have Food-101 [8] which consists of 101 food categories, with 101,000 images in total and 1,000 images for each category. In addition, the training images were not cleaned, and thus still contain some amount of noise. This comes mostly in the form of intense colors and sometimes wrong labels. All images were rescaled to have a maximum side length of 512 pixels.

The target dataset is downloaded directly from website https://www.floydhub.com/jean72human/datasets/rice-dataset called Rice-dataset. The raw dataset is splitted into training data of 348 images and test data of 80 images. This dataset may represent the small scalar of dataset we obtained in the daily appliaction.

B. Basic Model

We designed our feature extractor according to VGG-19, which is shown in figure 2. VGG-19 is a convolutional neural network that uses small 3×3 filters in all convolutional layers and has a total of 138M parameters. It is trained on more than a million images from the ImageNet database.

The network is 19 layers deep and can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224×224.

C. Problem Formulation

Consider a two source domain adaptation problem with source domains $D_s = \{A_n, B_n\}$, we also have train target domain C_n, and test target domain D_n. The multi-sources domain adaption problems aims to find a hypothesis in the given hypothesis space H, which minimizes the testing target error on D_n.

D. Our Framework

To achieve this goal, we propose our framework as shown in figure 1. It is composed of feature extractors, feature concatenation, and classifiers based on this [9]. Our methods can be summarized as followed, which is shown in figure1. The training process contains the following 4 main steps :

1) Train the models $E1$, $E2$ based on Food-101 and Food-256 respectively by using VGG-19.
2) Share parameters of $E1$ and $E2$ in the feature extraction layers with target domain model E.
3) Take the rice training data as input, and stitch the feature extracted by VGG-19 in $E1$ and feature extracted by VGG-19 in $E2$ together.
4) Take this kind of mixture of the source domains as the input, then a target classifier is trained on this with cross-entropy loss.

The algorithm is demonstrated as follows.

Algorithm 1: Framework of transfer learning for our system

Input: The set of sample in dataset *Food-256* A_n; The set of samples in dataset *Food-101* B_n; A small batch of target training dataset Rice C_n; A small batch of target test dataset Rice D_n:

Output: Model of the target domain, E.

```
function A_n, B_n ... E:
    Training model E_1 on the A_n based on VGG − 19.
    Training model E_2 on the B_n based on VGG − 19.
    Sharing the parameters of E_1, E_2 in feature
      extractors with E.
    Extracting features T_1 and T_2 of C_n using model
      E.
    Training model E on the mixture of T_1 and T_2.
    Classifying samples in D_n using model E.
    return E and predictions of D_n.
end function
```

IV. EXPERIMENTS

A. Experimental setup

In order to evaluate the effectiveness of our proposed method, we perform three experiments based on three food classification datasets: Food-101, Food-256 and a simple Rice-dataset. Food-101 dataset contains 101 food categories, with 101,000 images in total and 1000 for each category. Food-256 contains 256 food categories. Rice-dataset, however, contains only 4 kinds of rice images: fried rice, jollof, plain rice, and waakye. Example images from dataset can be found in Figure 3.

For Food-101 dataset, we have 101000 images in total, and we split them into 100,000 images as training set and 1,000 images as testing set. For Food-256 dataset, we have 31,395 images in total and we split them into 30,000 images as training set and 1395 images as testing set. Rice-dataset has

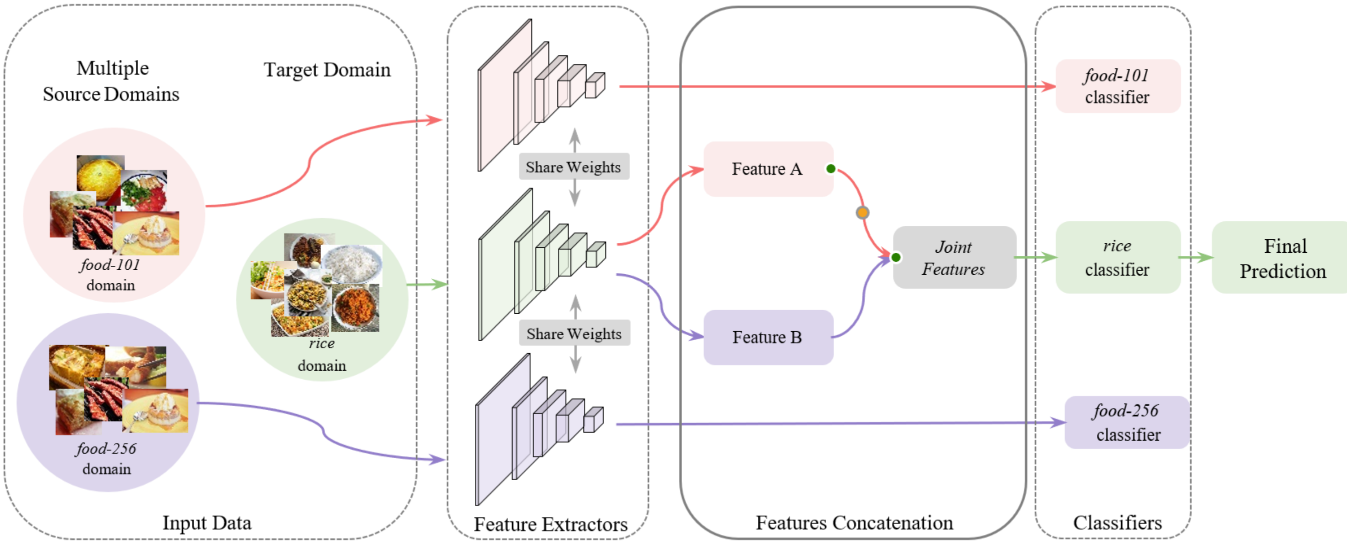

Fig. 1: Our feature transferable framework for Multi-source transfer learning

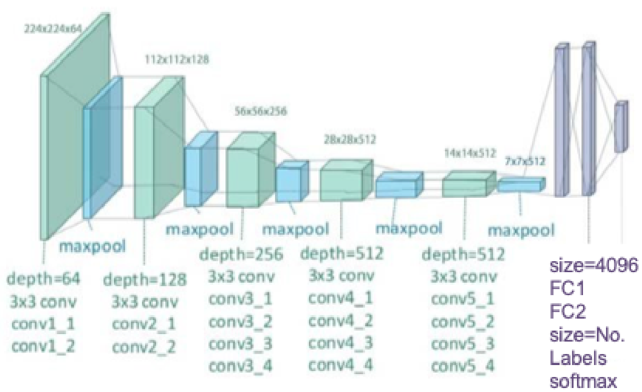

Fig. 2: The basic model of our framework [10]

only 428 images in total, thus we separated it into 348 images as training set and 80 images as testing set. We use VGG-19 architecture as our baseline neural network for both source and target task training and change the last layer of its classifier to adapt to number of categories in corresponding dataset. All of our experiments are implemented using Pytorch.

In the first experiment, we use the whole Food-101 and Food-256 dataset as our source domain and train on them respectively, ending up with pretrained feature extraction networks. Then we use the feature extractors trained on Food-101 and Food-256 as frozen feature extractor to further train on Rice-dataset, respectively. Using our proposed network, we then use the frozen feature extractors of Food-101 and Food-256 synchronously, concatenate the features of these two models together to further train on Rice-dataset.

In the second experiment, we use only 4 kinds of food in Food-101 and only 4 kinds of food in Food-256 dataset as our source domain, train on them respectively as the first experiment, ending up with pretrained feature extraction networks. The 4 kinds of food in Food-101 do not contain any categories similar to rice(apple pie, baby back ribs, baklava and beef carpaccio), while the 4 kinds of food in Food-256 are all similar with rice(rice, eels on rice, pilaf and chicken and egg on rice).

In the third experiment, we also use 4 kinds of food in Food-101 and only 4 kinds of food in Food-256 dataset as our source domain. However, here the 4 kinds of food in Food-256 contains no food about rice (spaghetti, takoyaki, waffle, and kung pao chicken). Also, since there are only 498 compared with 4000 images in 4 kinds of categories from Food-101, we sample 400 images in 4000 to make the amount of data similar in such two source domain datasets. Thus, this becomes a problem with relatively small amount of training and testing examples.

We also obtained the baseline performance of training and testing on pure Rice-dataset using VGG-19 architecture in order to compare with our multi-source methods.

B. Results on Food-101 and Food-256 dataset

In this part, we use the whole dataset of Food-101 and Food-256 as source domains. The Food-101 dataset is a collection of 101 kinds of food with 101,000 images in total, and we use 100,000 of them as training samples while the rest 1000 as testing set. We train a feature extractor based on Food-101 dataset. For Food-256, we use 30,000 images as training set and 1395 images as testing dataset. We train the network using number of epochs = 10. Then we transfer from Food-101 and Food-256 to Rice-dataset respectively by fixing the feature extractors and train a classifier on Rice-dataset. Finally, we use Food-101 and Food-256 as two source domains and transfer them to Rice-dataset . The results of single source to target domain are over 30 epochs and the result of multi-source is over 60 epochs. The result of Rice-dataset trained from scrach comes from using only the Rice-dataset as the source

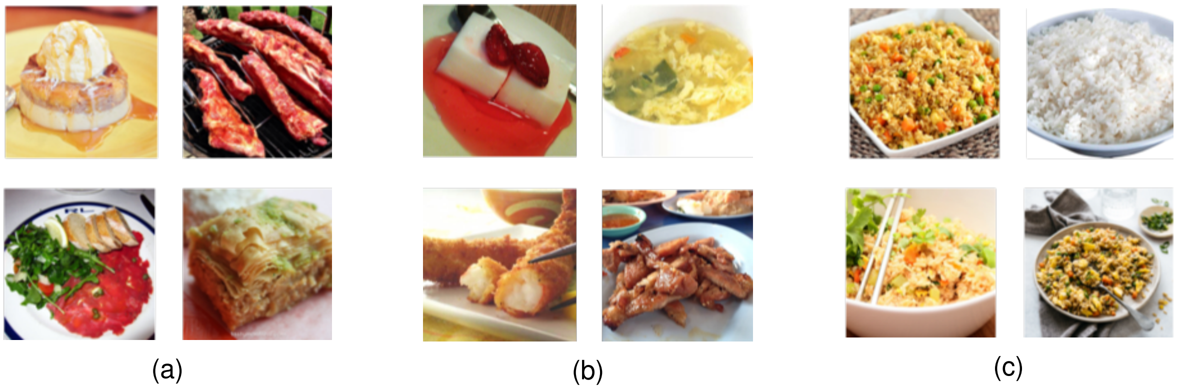

Fig. 3: Examples images from the (a) Food-101, (b) Food-256, and (c) Rice-dataset.

domain and training for 60 epochs. For a fair comparison, all the experiments are based on VGG-19 pretrained architecture.

The results are shown in Table I. Accuracy of testing on Food-101 and Food-256 models are not high for that they are trained only 10 epochs which may not be enough for the large amount of data in original Food-101 and Food-256. Our model of multi-source achieves an 85.00% average accuracy, outperforming results of transfering from Food-101 and Food-256 to Rice-dataset by 4.25% and 3.25%, respectively. One interesting observation is that the results of our proposed multi-source model is lower than that of training Rice-dataset from scratch. This phenomenon is probably due to the fact that the target dataset here is sufficient to build a model, and we can always get the perfect classifier when training and testing on the same domain or dataset with similarity.

models	train accuracy	test accuracy	epoch
rice dataset only	99.85±0.15	88.75±1.25	60
food-101 only	50.36±1.22	49.93±0.77	10
food-256 only	35.35±1.61	35.25±2.53	10
food-101 → rice	99.71±0.29	80.75±3.25	30
food-256 → rice	99.73±0.27	81.75±5.75	30
food-101+256 → rice	97.70±2.29	85.00±2.50	30

TABLE I: Results of complete 101 and 256 categories (Experimental results for using complete Food-101 and Food-256 as source domains. Our proposed multi-source domain adaptation method achieves 85.00% accuracy, outperforming the single source methods by 4.25%.)

C. Results on subsets of Food-101 and Food-256 dataset

Since the Food-101 and Food-256 may have too much categories compared with Rice-dataset (101 and 256 vs. 4), and the amount of data may be too much in comparison with Rice-dataset (101,000 and 31,395 vs. 428), we consider that we should make the amount of data balanced. We randomly construct a set of 4 categories from Food-101 and Food-256 using the first 4 kinds of food, where all the 4 kinds of food from Food-256 are rice images, and the results are shown in Table II. Also, Table III is especially experimented on 4 kinds of categories in Food-101 and 4 kinds of food in Food-256 without any similar rice images. The results of single source to target domain are over 30 epochs and the results of multi-source are over 30 or 60 epochs. From Table II, we can easily find out that the test accuracy of multi-source model (87.50%) is close to the one that is trained on Rice-dataset only (88.75%) and slightly higher than results of single source domain adaptation, which is 87.43% and 86.75%. Similar results can be found in Table III. Furthermore, the result classification accuracy illustrates that there exists over-fitting problems, since the train accuracy is nearly 100 percent, which is higher than the test accuracy, though we do improve the code by reducing the numbers of features and so on.

Comparing the experimental results of the subsets with and without rice, an interesting phenomenon can be discovered that the performance of transferring with rice (87.50%) is higher than that without rice (87.08%). This can be illustrated that since we intentionally select some rice data into training set, it improves the performance. Correlation between different rices food (such as fried rice, black rice and porridge to white rice) is apparently higher than that between rices and other kinds of food (such as desserts, soup and fried noodles). Actually, the difference between two subsets is smaller and both are close to the common one without task transfer, convincing a marvelous success of our framework.

models	train accuracy	test accuracy	epoch
rice dataset only	99.85±0.15	88.75±1.25	60
4-in-food-101 only	97.64±0.68	85.50±1.25	10
4-in-food-256 only	99.88±0.51	94.09±1.07	10
4-in-food-101 → rice	99.71±0.29	86.75±3.00	30
4-in-food-256 → rice	99.42±0.58	87.43±1.25	30
4-infood-101+256 → rice	98.56±1.44	87.50±2.50	30

TABLE II: Results of 4 categories in 101 and 4 categories in 256 (Experimental results for using subsets of Food-101 and Food-256 with similar rice samples as source domains. Our proposed multi-source domain adaptation method achieves 87.50% accuracy, slightly outperforming the single source methods.)

With results of the above three experiments, we find that our proposed multi-source method works more efficiently when using the whole Food-101 and Food-256 datasets as source domains. Results of subsets of Food-101 and Food-256 show

models	train accuracy	test accuracy	epoch
rice dataset only	99.85±0.15	88.75±1.25	60
4-in-food-101 only	99.44±0.56	83.75±2.25	30
4-in-food-256 only	99.75±0.25	96.94±1.03	30
4-in-food-101 → rice	99.43±0.57	86.42±2.32	60
4-in-food-256 → rice	99.56±0.15	86.66±1.15	60
4-infood-101+256 → rice	99.14±0.86	87.08±1.25	60

TABLE III: Results of 4 categories in 101 and 4 categories in 256 (Experimental results for using subsets of Food-101 and Food-256 without similar rice samples as source domains. Our proposed multi-source domain adaptation method achieves 87.08% accuracy, slightly outperforming the single source methods.)

that multi-source method performs slightly better than that of single source methods, illustrating that there is still a large space for improvement when learning from limited amount of data. In addition, the results show that using more correlated data with target in source domains for pretraining feature extractors will have positive effects on transfering.

V. Conclusion

In this paper, we presented our multi-source domain adaptation method inspired by the fact that we may have access to a few pre-trained neural networks or feature extractors on source training data, instead of the original data. By combining features from multiple pre-trained feature extractors, we were able to make full use of the given source domain knowledge and adapt to a target domain. Results of our experiments showed that this method was more efficient when source feature extractors were pre-trained on sufficient amount of data, which illustrates the importance of big data. Also, better performance of using source domains that were more similar with the target domain illustrated that transferring between similar domains may be more helpful to leverage the differences among domains.

In the future, we can further expand our work from two source domains to more than three domains. Also, we may have better improvement on how to combine different features or feature extractors of multiple different source domains. In addition, we hope to measure the transferabiity of different sources, or use quantitative representation to measure the correlation between domains, like HGR maximal correlations and so on, to learn more related and important information for domain adaptation.

References

[1] J. Li, Z. Xu, Y. Wong, Q. Zhao, and M. S. Kankanhalli, "Gradmix: Multi-source transfer across domains and tasks," 2018.

[2] Q. Sun, R. Chattopadhyay, S. Panchanathan, and J. Ye, "A two-stage weighting framework for multi-source domain adaptation," in *Advances in neural information processing systems*, 2011, pp. 505–513.

[3] J. He, D. C. Gondek, R. D. Lawrence, and E. C. Vijil, "Graph-based framework for multi-task multi-view learning," Mar. 24 2015, uS Patent 8,990,128.

[4] Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 1855–1862.

[5] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 361–368.

[6] Z. Ding, H. Zhao, and Y. Fu, "Multi-source transfer learning," in *Learning Representation for Multi-View Data Analysis*. Springer, 2019, pp. 175–202.

[7] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," in *Proc. of ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV)*, 2014.

[8] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – mining discriminative components with random forests," in *European Conference on Computer Vision*, 2014.

[9] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1406–1415.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.