Sparse Deep Transfer Learning for Convolutional Neural Network

Jiaming Liu,[*1] **Yali Wang,**[*1] **Yu Qiao**[†1,2]
[1]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[2]The Chinese University of Hong Kong, Hong Kong
jiaming.liu@email.ucr.edu, {yl.wang, yu.qiao}@siat.ac.cn

Abstract

Extensive studies have demonstrated that the representations of convolutional neural networks (CNN), which are learned from a large-scale data set in the source domain, can be effectively transferred to a new target domain. However, compared to the source domain, the target domain often has limited data in practice. In this case, overfitting may significantly depress transferability, due to the model redundancy of the intensive CNN structures. To deal with this difficulty, we propose a novel sparse deep transfer learning approach for CNN. There are three main contributions in this work. First, we introduce a Sparse-SourceNet to reduce the redundancy in the source domain. Second, we introduce a Hybrid-TransferNet to improve the generalization ability and the prediction accuracy of transfer learning, by taking advantage of both model sparsity and implicit knowledge. Third, we introduce a Sparse-TargetNet, where we prune our Hybrid-TransferNet to obtain a highly-compact, source-knowledge-integrated CNN in the target domain. To examine the effectiveness of our methods, we perform our sparse deep transfer learning approach on a number of benchmark transfer learning tasks. The results show that, compared to the standard fine-tuning approach, our proposed approach achieves a significant pruning rate on CNN while improves the accuracy of transfer learning.

1 Introduction

Over the past few years, convolutional neural networks (CNN) (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2014b; Szegedy et al. 2015; He et al. 2015) have achieved remarkable successes in a number of large-scale computer vision tasks (Krizhevsky, Sutskever, and Hinton 2012; Sun et al. 2014; Zhou et al. 2014; Simonyan and Zisserman 2014a). Extensive studies have shown that the representations of CNN, which are learned from a large-scale data set in the source domain, can be effectively transferred to a new target domain (Yosinski et al. 2014; Sharif Razavian et al. 2014; Donahue et al. 2014; Azizpour et al. 2015). However, compared to the source domain, the target domain often has limited data in practice. In this case, the transferring process may suffer from overfitting, due to the model redundancy of the intensive CNN structures.

In this paper, we introduce a novel sparse deep transfer learning to handle this difficulty, by taking advantage of both deep model compression and knowledge transferring. Specifically, we make three main contributions as follows. **First**, we introduce a *Sparse-SourceNet* by pruning CNN in the source domain. This allows to reduce the redundancy of CNN in the source domain, and subsequently alleviate the risk of overfitting in the next transferring process. **Second**, we propose a *Hybrid-TransferNet* in which an extra branch is incorporated into the modified *Sparse-SourceNet*. With this extra branch, our *Hybrid-TransferNet* can effectively exploit implicit source-domain knowledge during transferring. More importantly, due to the model sparsity and the implicit knowledge, our *Hybrid-TransferNet* can improve the generalization ability and the prediction accuracy of transfer learning. **Third**, we develop a *Sparse-TargetNet*, by further pruning our *Hybrid-TransferNet* in the target domain. This design helps to reduce the redundancy of the transferred model to obtain a highly-compact, source-knowledge-integrated CNN for the target domain. To show effectiveness, we perform our sparse deep transfer learning approach on a number of challenging transfer learning tasks for the famous AlexNet and VGGNet, namely, transferring AlexNet from object recognition with ImageNet to scene recognition with MIT Indoor67 or fine-grained flower recognition with Flower102, and transferring VGGNet for human action recognition in videos, from UCF101 to HMDB51. All results demonstrate that, compared to the standard fine-tuning approach, our approach achieves a significant pruning rate on CNN while improves accuracy of transfer learning.

2 Related Works

Transfer Learning for CNN: Transfer learning has been widely used for CNN, since CNN which is trained on a large-scale source data set, can be applied to a new target data set (Yosinski et al. 2014; Sharif Razavian et al. 2014; Donahue et al. 2014; Azizpour et al. 2015). One approach is to extract features of the target data directly from the source-data-pretrained CNN, and then use these features to train a linear classifier (such as SVM) for prediction (Sharif Razavian et al. 2014; Donahue et al. 2014). However, the performance of this approach is often restricted, due to the domain difference between source and target. Alternatively, fine-tuning the source-data-pretrained CNN with the target data

[*] Equally-contributed first authors

[†] Corresponding author

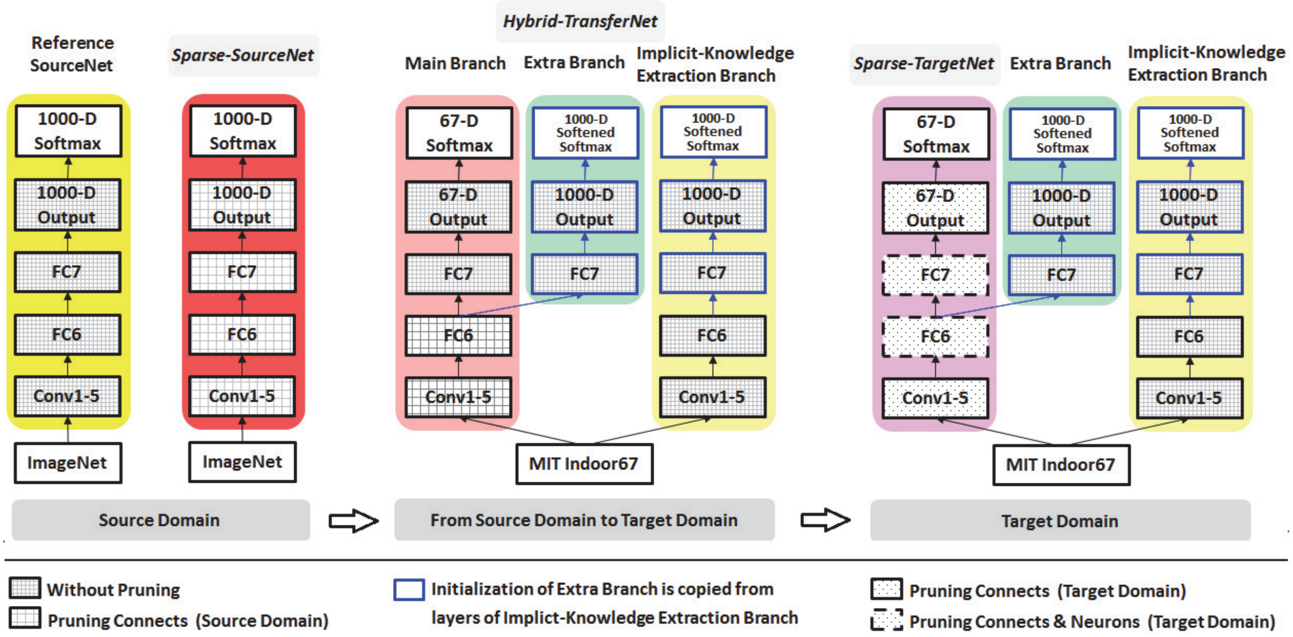

Reference SourceNet
Sparse-SourceNet
Hybrid-TransferNet
Main Branch
Extra Branch
Implicit-Knowledge Extraction Branch
Sparse-TargetNet
Extra Branch
Implicit-Knowledge Extraction Branch
1000-D Softmax
1000-D Output
FC7
FC6
Conv1-5
ImageNet
67-D Softmax
67-D Output
1000-D Softened Softmax
MIT Indoor67
Source Domain
From Source Domain to Target Domain
Target Domain
Without Pruning
Pruning Connects (Source Domain)
Initialization of Extra Branch is copied from layers of Implict-Knowledge Extraction Branch
Pruning Connects (Target Domain)
Pruning Connects & Neurons (Target Domain)

Figure 1: An illustration of our sparse deep transfer learning approach for AlexNet (8-layer CNN), from ImageNet (source, 1000 object classes) to MIT Indoor67 (target, 67 scene classes). **In the source domain**, we introduce a *Sparse-SourceNet* by pruning connections of *Reference-SourceNet*, in order to reduce model redundancy in the source domain. **From source to target**, we design a *Hybrid-TransferNet* which consists of main branch, implicit-knowledge-extraction branch, and extra branch. The main branch is *Sparse-SourceNet* with target-domain-related modifications. The sparsity of this branch, which inherits from *Sparse-SourceNet*, can alleviate overfitting for transfer learning. The implicit-knowledge-extraction branch is *Reference-SourceNet* with a softened softmax output. It is used to generate the implicit source-domain knowledge for the target domain. The extra branch is initialized by copying the corresponding layers of the implicit-knowledge-extraction branch. It is used to incorporate the implicit knowledge to assist transfer learning. **In the target domain**, we propose a *Sparse-TargetNet* by pruning connections and neurons of the main branch with implicit knowledge. It is a highly-compact, source-knowledge-integrated CNN for target. More details can be found in Section 3 and our experiments.

can alleviate the domain difference (Yosinski et al. 2014). But this approach often falls into overfitting, when CNN is intensive and the size of the target data is limited.

Deep Model Compression: Modern deep neural networks share a widespread property: redundancy. To reduce redundancy, deep learning researchers have made efforts to compress deep models by low-rank decomposition (Zhang et al. 2015; Denton et al. 2014; Jaderberg, Vedaldi, and Zisserman 2014), distilling knowledge from cumbersome models to small models (Hinton, Vinyals, and Dean 2015; Romero et al. 2014), pruning connections (Han et al. 2015; Han, Mao, and Dally 2015) and so on. In particular, pruning connections in (Han et al. 2015; Han, Mao, and Dally 2015) provides a simple but effective strategy for redundancy reduction, where a high pruning rate can be achieved with little loss of accuracy. However, all these approaches compress deep models on the same domain.

Different from all the methods above, we propose a novel sparse deep transfer learning approach for CNN in this paper, where we take advantage of both deep model compression and knowledge transferring to reduce overfitting and improve accuracy of transfer learning, especially when the size of target data is limited.

3 Sparse Deep Transfer Learning for CNN

In this section, we introduce the proposed sparse deep transfer learning approach for CNN, in order to reduce overfitting and improve accuracy for transferring CNN from the large-scale source data to the limited target data. An illustration of our approach is shown in Fig. 1.

3.1 Source Domain: Sparse-SourceNet

Firstly, we propose to sparsify the intensive CNN which is trained on the large-scale source data, before we transfer CNN from source to target. This is mainly due to the fact that redundancy in the source-domain-related CNN may lead to overfitting during transfer learning, especially when the size of target data is limited (Yosinski et al. 2014).

Hence, we exploit the iterative pruning network strategy in (Han et al. 2015) to reduce redundancy in the source domain, where the connections with low weights are proportionally removed in the prune-phase, and the remaining weights are fine-tuned with the source data in the retrain-phase. We denote the original CNN as *Reference-SourceNet*, and denote the pruned CNN as our *Sparse-SourceNet*. Note that, our contribution is not the pruning technique itself but

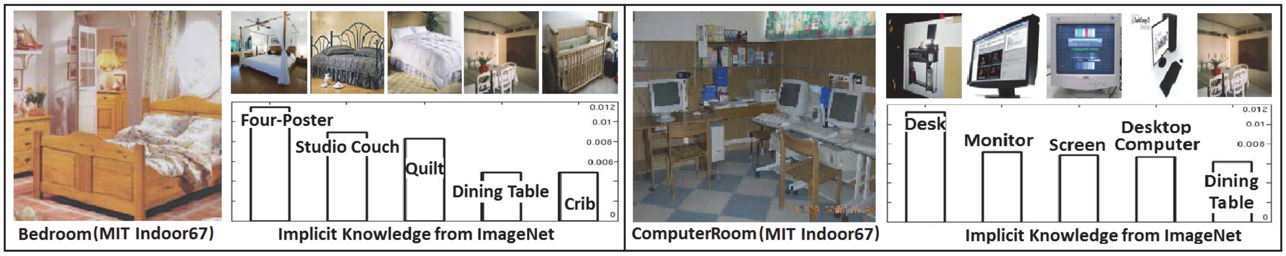


Figure 2: An illustration of implicit knowledge from source (ImageNet). This knowledge is the softened output of the reference AlexNet. Here, we feed 50 images in the Bedroom / ComputerRoom class of MIT Indoor67 to the reference AlexNet, and then compute the mean of the softened outputs over these images as an illustration. We show the top-5 object classes from ImageNet (source) which are most likely to appear in the Bedroom / ComputerRoom class of MIT Indoor67 (target).

the fact that the sparsity of our *Sparse-SourceNet* can help to reduce overfitting for transfer learning in the next stage.

3.2 From Source to Target: Hybrid-TransferNet

Based on our *Sparse-SourceNet*, we propose a novel *Hybrid-TransferNet* to introduce the implicit source-domain knowledge to assist transfer learning. This is mainly motivated by the fact that the relations between source and target can provide extra important knowledge about the target domain. Suppose that source is object (ImageNet) and target is scene (MIT Indoor67), as shown in Fig. 2. To better understand a scene image in the target domain, it is often crucial to leverage the implicit knowledge from the source domain, such as which classes of objects are likely to appear in this scene. Hence, to extract the implicit knowledge from the source domain and incorporate it into transfer learning, we design the following *Hybrid-TransferNet* with the main branch, the implicit-knowledge-extraction branch and the extra branch.

(I) Main Branch is used to make prediction for target. It is our *Sparse-SourceNet* but with the target-domain-related modifications in (Yosinski et al. 2014). First, we change the output layer of *Sparse-SourceNet* to the target classes, in order to perform transfer learning from source to target. Furthermore, we propose to recover the top N_m layers of *Sparse-SourceNet* to be dense and re-initialize these layers randomly to increase transferability.

(II) Implicit-Knowledge-Extraction Branch is used to extract the implicit source-domain knowledge. Specifically, we feed the target image into *Reference-SourceNet* and obtain the softmax output $\mathbf{p}_R = Softmax(\mathbf{a}_R)$, where $\mathbf{a}_R$ is the pre-softmax activation vector. Note that, the softmax output $\mathbf{p}_R$ reflects the implicit source-domain knowledge for the target domain, since it contains extra information such as which classes in the source domain are important for recognition in the target domain. In this work, we use the softened version of $\mathbf{p}_R$, i.e., $\mathbf{p}_R^\tau$ with a temperature ($\tau > 1$), as the implicit knowledge from the source domain, because

$$\mathbf{p}_R^\tau = Softmax(\frac{\mathbf{a}_R}{\tau}), \tag{1}$$

often contains richer information than $\mathbf{p}_R$ (Hinton, Vinyals, and Dean 2015; Romero et al. 2014).

(III) Extra Branch is used to incorporate the implicit source-domain knowledge into transfer learning. First, we copy the top N_e layers of the implicit-knowledge-extraction branch as the initialization of our extra branch. This design is motivated by the fact that, high-level features of the implicit-knowledge-extraction branch can be used as a hint to guide the learning of our extra branch. Next, to use the implicit source-domain knowledge, we soften the softmax output of our extra branch with the same temperature τ in Eq. (1),

$$\mathbf{p}_{extra}^\tau = Softmax(\frac{\mathbf{a}_{extra}}{\tau}), \tag{2}$$

where $\mathbf{a}_{extra}$ is the pre-softmax activation vector. Finally, we add this extra branch on top of the $(N - N_e)$-th layer of our main branch, in order to integrate the implicit knowledge to assist transfer learning.

(IV) Hybrid-TransferNet Training is performed by using the following total loss,

$$\mathcal{L}_{total} = \mathcal{L}_{main} + \lambda \mathcal{L}_{extra}, \tag{3}$$

where λ is a trade-off weight.

The main loss $\mathcal{L}_{main}$ is the cross entropy between the softmax output of our main branch $\mathbf{p}_{main}$ and the target label $\mathbf{y}_{target}$,

$$\mathcal{L}_{main} = CrossEntropy(\mathbf{p}_{main}, \mathbf{y}_{target}). \tag{4}$$

Note that, **our main branch is sparse because it is originally from our *Sparse-SourceNet*.** As a result, **the sparsity of our main branch can alleviate overfitting for transfer learning via $\mathcal{L}_{main}$** .

The extra loss $\mathcal{L}_{extra}$ is the cross entropy between the softened softmax output of our extra branch $\mathbf{p}_{extra}^\tau$ in Eq. (2) and the implicit source-domain knowledge $\mathbf{p}_R^\tau$ in Eq. (1),

$$\mathcal{L}_{extra} = CrossEntropy(\mathbf{p}_{extra}^\tau, \mathbf{p}_R^\tau). \tag{5}$$

With this loss, one can use our extra branch to integrate the knowledge extracted from our implicit-knowledge-extraction branch into the transferring procedure. Note that, this implicit knowledge can inherit the regularization merit in (Hinton, Vinyals, and Dean 2015), i.e., the generalization ability of our cumbersome implicit-knowledge-extraction branch is transferred to our sparse main branch via our extra branch. More importantly, the source domain and the target domain are different in our case. Hence, different from (Hinton, Vinyals, and Dean 2015), **our implicit knowledge provides extra important information of the source domain to improve the prediction accuracy in the target domain.**

3.3 Target Domain: Sparse-TargetNet

After transferring from source to target with our *Hybrid-TransferNet*, we propose to reduce redundancy in the target domain. This is because the top N_m layers in the main branch of our *Hybrid-TransferNet* are recovered to be dense to increase transferability. After the domain is transferred to target, these layers may be redundant. Additionally, compared to the large source data set, the target data set is limited. In this case, **many connections and even neurons may not be useful after transferring from source to target**.

Since our main branch of *Hybrid-TransferNet* is used for prediction in the target domain, we propose to prune the main branch of our *Hybrid-TransferNet* as follows. **First**, we use the iterative pruning network strategy in (Han et al. 2015) to prune connections in our main branch of *Hybrid-TransferNet*. But different from (Han et al. 2015), the implicit source-domain knowledge is exploited in our case. Hence, after performing the prune-phase of (Han et al. 2015) at each iteration, we use our total loss in Eq. (3) to retrain the remaining connections, in order to leverage the extra knowledge to reduce overfitting in the target domain.

Second, we propose to prune neurons of the inner-product layers in our main branch, due to the fact that many high-level features (neurons) of the transferred model may not be quite useful in the target domain, especially when the size of the target data is limited. Specifically, for an inner-product layer to be pruned, we compute the mean of the activation vectors over a data subset (randomly chosen from the training set), and proportionally delete the low-activated neurons. Then, we perform re-training with our total loss in Eq. (3). The resulting pruned main branch is denoted as ***Sparse-TargetNet*** which is **a highly-compact, source-knowledge-integrated CNN for the target domain.**

4 Experiments

To examine the effectiveness of the proposed methods, we perform our sparse deep transfer learning approach for two benchmark CNNs, i.e., 8-layer AlexNet (Krizhevsky, Sutskever, and Hinton 2012) and 16-layer VGGNet (Simonyan and Zisserman 2014b).

4.1 Sparse Deep Transfer Learning for AlexNet

For AlexNet, we evaluate our approach on two popular transfer learning tasks (Sharif Razavian et al. 2014; Azizpour et al. 2015), where the source domain of both tasks is object recognition with ImageNet ILSVRC-2012 (1000 object classes, $>$ 1 million images) (Deng et al. 2009). The target domains are respectively scene recognition with MIT Indoor67 (67 scene classes, 15,620 images) (Quattoni and Torralba 2009), and fine-grained flower recognition with Flower102 (102 flower classes, 7,169 images) (Nilsback and Zisserman 2008). Note that, the domain difference between ImageNet and MIT Indoor67 / Flower102 is relatively large (Azizpour et al. 2015), and the training sizes of these target data are quite limited (each class of MIT Indoor67 / Flower102: around 100/10). Hence, these two transfer learning tasks are challenging but reasonable choices to validate our proposed approach.

(I) Source Domain: Sparse-SourceNet To make our results extensible, we choose AlexNet (trained by ImageNet) from Caffe model zoo (Jia et al. 2014) as our *Reference-SourceNet*. Then, we perform the iterative pruning strategy (Han et al. 2015) on our *Reference-SourceNet* to reduce model redundancy in the source domain. As a result, we obtain our *Sparse-SourceNet* which achieves a comparable result to (Han et al. 2015), i.e., total pruning rate (ours vs (Han et al. 2015)) is 88.0% vs 89.0%, top-1 accuracy (ours vs (Han et al. 2015)) is 57.4% vs 57.2%, top-5 accuracy (ours vs (Han et al. 2015)) is 80.4% vs 80.3%. Note that, our contribution on *Sparse-SourceNet* is not the pruning strategy itself but the fact that its model sparsity can reduce overfitting for transfer learning. Hence, we next design our *Hybrid-TransferNet* to show if our *Sparse-SourceNet* can help to improve transfer learning.

(II) From Source to Target: Hybrid-TransferNet We firstly introduce the settings of *Hybrid-TransferNet*. An illustration is shown in Fig. 1. For the main branch, we change the output layer of *Sparse-SourceNet* to the target classes (MIT Indoor67 or Flower102). Then we recover FC7, the output layer to be dense and re-initialize them randomly to increase transferability. Note that, although FC7 and the output layer are dense, other layers still inherit sparsity from *Sparse-SourceNet*. Hence, our main branch is sparse. For the implicit-knowledge-extraction branch, we feed target images (MIT Indoor67 or Flower102) into *Reference-SourceNet*, and output the implicit knowledge by the softened softmax (τ is four). For the extra branch, we initialize it by copying FC7 & the output layer of implicit-knowledge-extraction branch. Finally, we train *Hybrid-TransferNet* by the total loss in Eq. (3). The weight λ is one, so that the value of $\lambda\mathcal{L}_{extra}$ is set as about $0.1\mathcal{L}_{main}$ in Eq. (3). In this case, implicit knowledge positively enhances the training process via the extra loss without disturbing the main loss negatively.

We next evaluate **if model sparsity and implicit knowledge in our *Hybrid-TransferNet* can improve transfer learning**. To achieve this, we compare our *Hybrid-TransferNet* with the following baselines. The first baseline is *Hybrid-TransferNet* without both model sparsity and implicit knowledge, where there is only the main branch in *Hybrid-TransferNet* and this branch is initialized from the dense *Reference-SourceNet*, instead of *Sparse-SourceNet*. In fact, this is the net which is used in the standard fine-tuning approach of transfer learning. Hence we denote it as *Standard-TransferNet*. The second baseline is *Hybrid-TransferNet* with model sparsity but without implicit knowledge, where there is only the main branch in *Hybrid-TransferNet* and this branch is the sparse one which is originally from *Sparse-SourceNet*, as mentioned in our settings of *Hybrid-TransferNet* before. We denote it as *Hybrid-TransferNet(S+K-)*. The third baseline is *Hybrid-TransferNet* with implicit knowledge but without model sparsity, where all branches exist in *Hybrid-TransferNet*, but the main branch is dense and same as *Standard-TransferNet*. We denote it as *Hybrid-TransferNet(S-K+)*.

In Table 1, we can see *Hybrid-TransferNet(S+K-)* outperforms *Standard-TransferNet*. This illustrates that **model sparsity of the main branch, which is originally from**

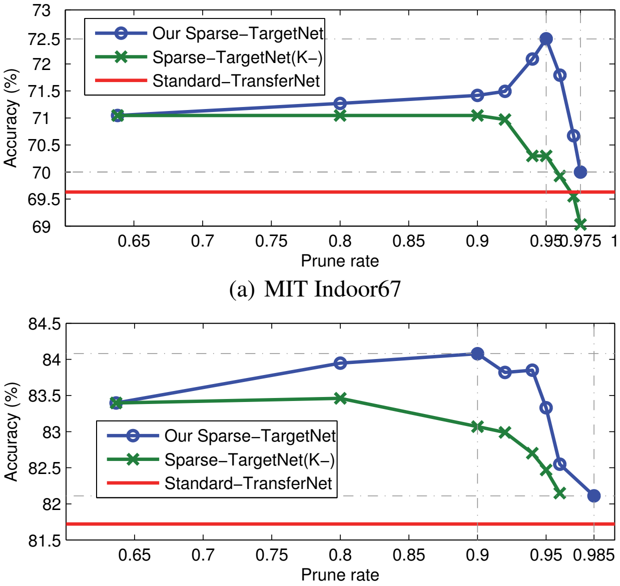
Table 1: Impact of model sparsity & implicit knowledge on the accuracy (ACC%) of transfer learning with our Hybrid-TransferNet (based on AlexNet, from ImageNet to MIT Indoor67 or Flower102). Standard-TransferNet is Hybrid-TransferNet without both model sparsity and implicit knowledge. Hybrid-TransferNet(S+K-) is Hybrid-TransferNet with model sparsity but without implicit knowledge. Hybrid-TransferNet(S-K+) is Hybrid-TransferNet without model sparsity but with implicit knowledge. Additionally, since the main branch is used to make prediction for target, ACC% is computed from the main branch of different transfer models.

ACC %	Indoor	Flower
Standard-TransferNet	69.6%	81.7%
Hybrid-TransferNet(S+K-)	70.1%	81.8%
Hybrid-TransferNet(S-K+)	69.9%	82.7%
our Hybrid-TransferNet	**71.0%**	**83.4%**

our *Sparse-SourceNet*, can alleviate overfitting for transfer learning. Next, *Hybrid-TransferNet(S-K+)* outperforms *Standard-TransferNet*. This indicates that **implicit knowledge provides extra source-domain information to improve accuracy for transfer learning**. Finally, our *Hybrid-TransferNet* takes advantage of both model sparsity and implicit knowledge, and it thus achieves the best accuracy among baselines when transferring from source to target.

(III) Target Domain: Sparse-TargetNet After transferring with our *Hybrid-TransferNet*, we propose to further reduce overfitting in the target domain. One important reason is that FC7 and the output layer in the main branch of our *Hybrid-TransferNet* are switched to be dense when transferring from source to target. This may introduce model redundancy in the target domain. Hence, we propose to prune the main branch of our *Hybrid-TransferNet* in the target domain and denote the resulting net as our *Sparse-TargetNet*.

Firstly, we **prune connections** with the strategy in Section 3.3. Different from (Han et al. 2015), we fine-tune the pruned model with our total loss (Eq. (3)) in the retraining phase, to incorporate implicit knowledge for prediction. We show accuracy & connection-pruning-rate curves of our *Sparse-TargetNet* in Fig. 3, where we compare our *Sparse-TargetNet* with *Sparse-TargetNet (K-)* and *Standard-TransferNet*. *Sparse-TargetNet (K-)* is the net by pruning the main branch of *Hybrid-TransferNet* without incorporating implicit knowledge, and *Standard-TransferNet* is the net for the standard fine-tuning approach of transfer learning. From Fig. 3, we can see that the beginning of *Sparse-TargetNet (K-)* and our *Sparse-TargetNet* is the same. This is because both nets start from the main branch of *Hybrid-TransferNet*, where the pruning rate for MIT Indoor67 / Flower102 is 63.8% / 63.6%. Furthermore, as we continue pruning connections in the target domain, the accuracy curve of our *Sparse-TargetNet* is consistently better than the one of *Sparse-TargetNet (K-)*. This illustrates that the implicit knowledge provides our *Sparse-TargetNet* with important source-domain supervision to improve accuracy in the target domain. Additionally, there is an accuracy-increasing procedure in the curve of our *Sparse-TargetNet*, which indicates that further pruning connections is helpful for overfitting reduction in the target domain. Finally, as expected, our *Sparse-TargetNet* significantly outperforms *Standard-TransferNet* in terms of both pruning rate and accuracy.



(a) MIT Indoor67

(b) Flower102

Figure 3: Accuracy & connection-pruning-rate curve of our Sparse-TargetNet. Sparse-TargetNet(K-): Sparse-TargetNet without implicit knowledge. Standard-TransferNet: standard fine-tuning for transfer learning. As there is no pruning in the Standard-TransferNet, its accuracy curve is a line.

Secondly, we **prune neurons** of FC6 and FC7 in our *Sparse-TargetNet*. This is mainly because high-level features from the transferred model are not quite important for the target domain, when the size of the target data is limited. Hence, we randomly choose 50 / 10 images per class from the training set of MIT Indoor67 / Flower102, and feed these images into our *Sparse-TargetNet* to compute the mean of activations (values after ReLU) for each neuron of FC6 and FC7. In each pruning iteration, we prune 512 lowest-activated neurons and perform retraining with our total loss in Eq. (3). This iterative procedure is stopped when the accuracy is the level of *Standard-TransferNet*. As shown in Table 2, almost half of the neurons in FC6 and FC7 are removed with little loss of accuracy.

In summary, we show the performance comparison of different models in Table 3. Compared to *Standard-TransferNet* for MIT Indoor67 / Flower102, our *Sparse-TargetNet* achieves a better accuracy with a large pruning rate, and obtains a significant pruning rate with little loss of accuracy. This indicates that **our *Sparse-TargetNet* is a highly-compact and accurate CNN in the target domain, taking advantage of both deep model compression and knowledge transferring**.

Table 2: Pruning neurons of FC6 and FC7 in our Sparse-TargetNet for the target domain (MIT Indoor67 / Flower102). Note that, when we prune neurons, the weights are automatically removed. After pruning almost half number of neurons in FC6 and FC7, the accuracy of our Sparse-TargetNet is 69.6% / 81.6%. It is with little loss of accuracy, compared to Standard-TransferNet (69.6% / 81.7%).

Indoor	Before/After Pruning		Prune %	
	Neurons	Weights	Neurons	Weights
FC6	4096/2048	433K/217K	50.0%	49.7%
FC7	4096/2560	144K/52K	37.5%	63.5%
Output	67/67	66K/44K	0.0%	33.4%
Total	8259/4675	644K/314K	43.4%	51.1%

Flower	Before/After Pruning		Prune %	
	Neurons	Weights	Neurons	Weights
FC6	4096/2048	145K/84K	50.0%	41.7%
FC7	4096/2048	50K/18K	50.0%	63.6%
Output	102/102	69K/38K	0.0%	45.3%
Total	8294/4198	264K/140K	49.4%	46.8%

Table 3: Summary of different models (based on Alexnet, transferring from ImageNet to MIT Indoor67 / Flower102). Sparse-TargetNet$_{C95}$ for MIT Indoor67 denotes our Sparse-TargetNet with 95% pruning rate after pruning connections in the target domain. Sparse-TargetNet$_{C97.5}$ for MIT Indoor67, Sparse-TargetNet$_{C90}$ and Sparse-TargetNet$_{C98.5}$ for Flower102 apply the similar notations. In addition, Sparse-TargetNet$_{N}$ is obtained by pruning neurons from Sparse-TargetNet$_{C97.5}$ / Sparse-TargetNet$_{C98.5}$ for MIT Indoor67 / Flower102. The accuracy (ACC%) and pruning rate (Prune %) are calculated by using the main branch of different models, as this branch is applied for recognition.

Indoor	ACC%	Prune %
Standard-TransferNet	69.6%	0.0%
our Hybrid-TransferNet	71.0%	63.8%
our Sparse-TargetNet$_{C95}$	**72.5**%	95.0%
our Sparse-TargetNet$_{C97.5}$	70.0%	97.5%
our Sparse-TargetNet$_{N}$	69.6%	**98.1**%

Flower	ACC%	Prune %
Standard-TransferNet	81.7%	0.0%
our Hybrid-TransferNet	83.4%	63.6%
our Sparse-TargetNet$_{C90}$	**84.1**%	90.0%
our Sparse-TargetNet$_{C98.5}$	82.1%	98.5%
our Sparse-TargetNet$_{N}$	81.6%	**98.7**%

Table 4: Summary of different models (based on VGGNet, transferring from UCF101 to HMDB51). As the main branch of different models is used for recognition in the target domain, ACC% and Prune% on HMDB51 (target) are calculated by using this branch. ACC% of two-stream is obtained by the output fusion of spatial and temporal streams.

Spatial Stream	ACC%	Prune %
Standard-TransferNet	43.4%	0.0%
our Hybrid-TransferNet	44.0%	78.2%
our Sparse-TargetNet	**44.1%**	**85.0%**

Temporal Stream	ACC%	Prune %
Standard-TransferNet	58.9%	0.0%
our Hybrid-TransferNet	59.0%	78.5%
our Sparse-TargetNet	**59.7%**	**85.0%**

Two-Stream	ACC%	Prune %
Standard-TransferNet	61.4%	0.0%
our Hybrid-TransferNet	61.4%	78.3%
our Sparse-TargetNet	**62.6%**	**85.0%**

4.2 Sparse Deep Transfer Learning for VGGNet

For 16-layer VGGNet, we evaluate our approach on a transfer learning task for human action recognition in the videos (Simonyan and Zisserman 2014a; Wang et al. 2015), where the source data set is UCF101 (101 action classes, 13,320 videos) (Soomro, Zamir, and Shah 2012), the target data set is HMBD51 (51 action classes, 6,849 videos) (Kuehne et al. 2013). This is a very challenging problem (Simonyan and Zisserman 2014a; Wang et al. 2015). One reason is that, compared with images, videos contain more complex high-level vision concepts and large intra-class variations which are difficult for recognition. More importantly, both UCF101 and HMDB51 are quite small, and overfitting can reduce the performance of transfer learning. Therefore, we choose this task to show effectiveness of our proposed approach.

To make our results extensible, we choose the benchmark two-stream CNNs (Simonyan and Zisserman 2014a; Wang et al. 2015) as our *Reference-SourceNet*, where the spatial / temporal VGGNets are respectively pre-trained with RGB images / stacked optical flow from UCF101 (Wang et al. 2015). We perform our sparse deep transfer learning for both VGGNets, by using the official first train/test split of both UCF101 and HMDB51. All the settings of our approach are the same as before, except that the basic structure is switched from AlexNet to VGGNet, the corresponding output layers are switched to the classes of UCF101 and HMDB51, the extra branch is added on the 14-th layer of VGGNet (an inner-product layer called FC6) for both streams, *Sparse-TargetNet* is obtained by only pruning connections in the target domain, due to the limited data in both UCF101 and HMDB51, λ is set as two for the total training loss in Eq. (3), and the proportion of spatial/temporal stream is one/four for output fusion of two-stream net.

We show the summary of our proposed approach in Table 4. For each stream, our *Hybrid-TransferNet* outperforms *Standard-TransferNet*. It illustrates that model sparsity and implicit knowledge from source can reduce overfitting for transfer learning. Furthermore, our *Sparse-TargetNet* outperforms our *Hybrid-TransferNet*, which indicates that pruning on the target domain can further alleviate overfitting and improve accuracy in the target domain. Finally, two-steam structure of our models is generally better than single-stream structure of our models, due to the spatial-temporal fusion.

5 Conclusion

In this work, we have proposed a novel sparse deep transfer learning approach for CNN. First, we pruned CNN in the source domain. The obtained *Sparse-SourceNet* reduces the source-domain-related redundancy and thus alleviate overfitting for transfer learning. Second, we proposed a *Hybrid-TransferNet*. It benefits from sparsity of *Sparse-SourceNet* and implicit knowledge to increase transferability from source to target. Finally, we developed a *Sparse-TargetNet* by pruning the main branch of *Hybrid-TransferNet* in the target domain to further reduce redundancy introduced by domain difference. Our experiments demonstrated that our approaches can achieve a significant pruning rate and improve accuracy on a number of transfer learning tasks for the benchmark AlexNet and VGGNet. In the future, it would be interesting to apply other compression strategies (Zhang et al. 2015; Denton et al. 2014; Jaderberg, Vedaldi, and Zisserman 2014) in our approach to further increase transferability of CNN.

6 Acknowledgments

This work is partly supported by National High-Tech Research and Development Program of China (2016YFC1400704, 2015AA042303), National Natural Science Foundation of China (61472410, 61502470), Guangdong Research Program (2014B050505017, 2015B010129013), and Shenzhen Research Program (JSGG20150925164740726, JCYJ20150925163005055, CXZZ20150930104115529, JCYJ20160229193541167).

References

Azizpour, H.; Razavian, A. S.; Sullivan, J.; Maki, A.; and Carlsson, S. 2015. Factors of transferability for a generic convnet representation. In *arXiv:1406.5774v3*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Denton, E.; Zaremba, W.; Bruna, J.; LeCun, Y.; and Fergus, R. 2014. Exploiting linear structure within convolutional networks for efficient evaluation. In *NIPS*.

Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; and Darrell, T. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*.

Han, S.; Pool, J.; Tran, J.; and Dally, W. J. 2015. Learning both weights and connections for efficient neural networks. In *NIPS*.

Han, S.; Mao, H.; and Dally, W. J. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *ICLR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep residual learning for image recognition. In *arXiv preprint arXiv:1512.03385*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a nueal network. In *arXiv:1503.02531*.

Jaderberg, M.; Vedaldi, A.; and Zisserman, A. 2014. Speeding up convolutional neural networks with low rank expansions. In *BMVC*.

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; and Darrell, T. 2014. Caffe: Convolutional architecture for fast feature embedding. In *arXiv preprint arXiv:1408.5093*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

Kuehne, H.; Jhuang, H.; Stiefelhagen, R.; and Serre, T. 2013. Hmdb51: A large video database for human motion recognition. In *High Performance Computing in Science and Engineering 12*. Springer. 571–582.

Nilsback, M.-E., and Zisserman, A. 2008. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*.

Quattoni, A., and Torralba, A. 2009. Recognizing indoor scenes. In *CVPR*.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2014. Fitnets: Hints for thin deep nets. In *arXiv:1412.6550v1*.

Sharif Razavian, A.; Azizpour, H.; Sullivan, J.; and Carlsson, S. 2014. Cnn features off-the-shelf: An astounding baseline for recognition. In *CVPR Workshops*.

Simonyan, K., and Zisserman, A. 2014a. Two-stream convolutional networks for action recognition in videos. In *NIPS*.

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

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Sun, Y.; Chen, Y.; Wang, X.; and Tang, X. 2014. Deep learning face representation by joint identification-verification. In *NIPS*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*.

Wang, L.; Xiong, Y.; Wang, Z.; and Qiao, Y. 2015. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*.

Yosinski, J.; Clune, J.; Bengio, Y.; and Lipson, H. 2014. How transferable are features in deep neural networks? In *NIPS*.

Zhang, X.; Zou, J.; He, K.; and Sun, J. 2015. Accelerating very deep convolutional networks for classification and detection. *IEEE T-PAMI*.

Zhou, B.; Lapedriza, A.; Xiao, J.; Torralba, A.; and Oliva, A. 2014. Learning deep features for scene recognition using places database. In *NIPS*.