Dense Residual Connection Finetuning

First Author[1] , **Second Author**[2] , **Third Author**[2,3] and **Fourth Author**[4]

[1]First Affiliation
[2]Second Affiliation
[3]Third Affiliation
[4]Fourth Affiliation

{first, second}@example.com, third@other.example.com, fourth@example.com

Abstract

Transfer learning is widely used for many applications, but has it's limitations: it is difficult to adapt intermediate layers of the network to a new learning task with less data available. However, with learning connection importance, models can better conform to new tasks. This paper presents a novel deep learning architecture, termed as DRCNet along with an unconventional technique of "connection-finetuning". Dense residual connection-finetuning is achievable through a strength parameter learned via backpropagation. In this research, we have shown that some connections are redundant and therefore, based on their strength, removing them can improve the overall performance. Results on multiple databases with intra (Same) and inter (Cross) experiments showcase the effectiveness of the proposed algorithm, for example, the cross-dataset experiments show improvement of 10-20% in classification accuracy compared to traditional ResNet architecture. Further, experiments also demonstrate that our novel method excels over existing approaches when limited data is available for training, thus reinforcing our claim.

1 Introduction

Since the beginning of deep learning, researchers have proposed several architectures to build robust and accurate learning models. Among all the Convolutional Neural Network (CNN) architectures, nowadays ResNet [He *et al.*, 2016] and DenseNet [Huang *et al.*, 2017] are widely used due to their superior performance and efficiency. These networks are not only used for the original problems for which they were designed, but a lot more research is going into how to modify these networks so that trained models can be adapted for other databases/tasks, also termed as Transfer Learning or Finetuning. In general, two methods have been used for model finetuning: (1) only the last few layers are trained and (2) the entire network is fine-tuned. Challenges have been observed with both the approaches: with the first option, intermediate layers do not receive any feedback while performing backpropagation for the target task, whereas the second method has a large number of parameters to train and therefore requires a substantial database as well. To solve this problem, architecture learning, hybrid architectures, automatic architecture selection and model finetuning have been proposed in literature. However, generalizability, scalability and performance for deeper architectures are still challenging tasks.

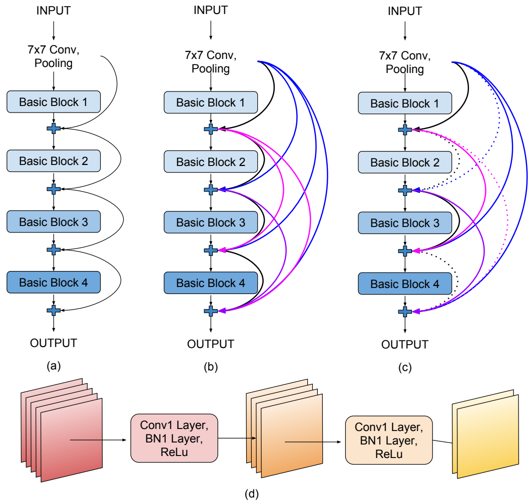

Figure 1: Illustration of the proposed architecture. (a) Skip-connections in a traditional ResNet architecture. (b) Skip-connections in proposed Densely-connected Residual Connection in ResNet architecture. We termed it as DRCNet. (c) Skip-connections after connection finetuning in DRCNet architecture, i.e. Sparse Connection Densely-connected Residual Net (SC-DRCNet) architecture (dashed lines show removed skip connections). (d) Structure of each Basic Block shown in parts (a) through (c).

In this research, we attempt to answer the following question: "Is there a better way to adapt deep learning models, such as ResNet, for different tasks without notably increasing the number of parameters required to train?". We take motivation from ResNet and DenseNet models in terms of their layout of connections, and design a new architecture, DRCNet. Figure 1 illustrates the concept and shows the relationship of a traditional ResNet architecture with the proposed DRCNet, and the proposed connection-finetuned DRCNet.

We observe that both ResNet and DenseNet models have skip connections: (1) in ResNet, there exists an identity mapping such that the output of an intermediate convolutional layer is connected with the immediately previous layer's output ($H(x) = f(x) + x$), and (2) in DenseNet, intermediate layers are connected densely to all the succeeding layers. Our hypothesis is that if the skip connections are introduced in the network intelligently, or learned during training of the architecture for a specific application, the network performance might improve, especially in the case of adapting the model from a different task. To model this concept, we propose *connection-finetuning* as a novel finetuning technique that modifies the skip connections while optimizing the model parameters.The effectiveness of the proposed DRCNet is demonstrated on multiple databases , including small sample size experiments as well.

2 Related Work

Considering that the proposed work focuses primarily on a novel approach for model adaptation, and to the best of our knowledge, no literature exists on connection-finetuning, we would like to draw attention to recent research advances in the field of Transfer learning.

Utilizing the transfer of information from the pre-trained model, Yosinski *et al.* [Yosinski *et al.*, 2014] have demonstrated that initialization using transferred features, along with finetuning, leads to improved performance on that target dataset with deep neural networks. Tajbaksh *et al.* [Tajbakhsh *et al.*, 2016] and Ahmed *et al.* [Ahmed *et al.*, 2017] have studied the performance of sufficiently finetuned deep networks for a variety of medical applications, ranging from radiology, cardiology to gastroenterology with tasks involving classification, detection, and segmentation. Similarly, Qu *et al.* [Qu *et al.*, 2019] have observed that the method of finetuning is mainly dependant on the selection of an appropriate source domain and have proposed a reinforcement learning algorithm to efficiently select source data and integrate it to a DNN-based Transfer Learning Model. It is observed that finetuning of a pretrained network often outperforms, if not is commensurate with training the same network from scratch or using handcrafted features.

Zhizhong *et al.* [Li and Hoiem, 2018] have proposed a novel Learning-without-Forgetting technique which uses only the training data of the target task to optimize new parameters, while preserving old parameters such that the performance on the original task remains unchanged.

Rebuffi *et al.* [Rebuffi *et al.*, 2017] [Rebuffi *et al.*, 2018] introduced the idea of adapter residual modules and universal parametric families that allow easy finetuning of deep networks to adapt to a multitude of different tasks and data, on the go. Rosenblum *et al.* [Rosenfeld and Tsotsos, 2018] have proposed Deep Adaptation Modules which constraints the new filters as simply linear combinations of existing ones. Guo *et al.* [Guo *et al.*, 2019] have proposed SpotTune algorithm to find the optimal finetuning strategy per instance for the target task and makes routing decisions based on a policy network. Mallya *et al.* [Mallya *et al.*, 2018] have proposed Piggyback networks which learn binary masks for each feature map in a pretrained network for adapting to another target task.

Along with finetuning of the model for specific applications, Casanova *et al.* [Casanova *et al.*, 2018] have modified the ResNet architecture while adding dense skip-connections. Similarly, Savarese *et al.* [Savarese *et al.*, 2016] proposed Gated ResNets wherein each skip connection is controlled by a gate (linear function on block output), activated by a scalar parameter. Hettinger *et al.* [Hettinger *et al.*, 2018] performed various experiments with a new architecture, replacing conventional blocks in ResNet with different Tandem Blocks, which are capable of learning any kind of mapping (including identity). The proposed dense residual connection finetuning method could also be considered a form of Neural Architecture Search (NAS). An example of the same, CondenseNets proposed by Huang *et al.* [Huang *et al.*, 2018], intelligently learns sparsified skip connection architectures and is relevant to our work. Their proposed method differs in the sense that it (1) compares with a baseline architecture of DenseNet, (2) learns a grouping of incoming features, (3) induces sparsity within connections of each group.

3 Proposed Method

In this section, we present the proposed DRCNet with connection-finetuning.

3.1 Connection Finetuning

In this research, a novel method has been proposed which intelligently learns the importance of a skip-connection and accordingly, ignores or preserves it. This has been termed as "connection-finetuning". In order to empirically evaluate the significance of the information being carried forward by any connection in a network, we propose to learn a Strength parameter, unique to each connection during training. Consider this as an indicator of a connection's contribution to the complete learning process. Finetuning connections refers to taking action, namely, retaining or dropping a connection from the network, in order to boost accuracy and robustness.

3.2 DRCNet

In order to evaluate the efficiency of connection-finetuning, we use ResNet architecture (which has skip-connections) as a baseline. Mathematically, skip-connections can be written as:

$$H(x) = f(x) + x \tag{1}$$

where, $H(x)$ is the output of a block of the network, x is the input to that block and function $f(x)$ represents the cumulative effect of the series of convolutions and batch normalization applied on the input during the forward pass in that block. A ResNet only considers the preceding output to be of importance to the learning process. To perform connection-finetuning, we first propose to modify ResNet framework by adding dense-connections. This means that the output of all prior blocks will be summed up with the convolution output of the current block. Here, dense-connections are added while following equation 2. The modified ResNet architecture is called as "DRCNet" as shown in figure 1.

For the n^{th} block in DRCNet, with each block having input $x_i, \forall i \in [1, n]$, the output of the n^{th} block is given as:

$$H(x_n) = f(x_n) + \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2}$$

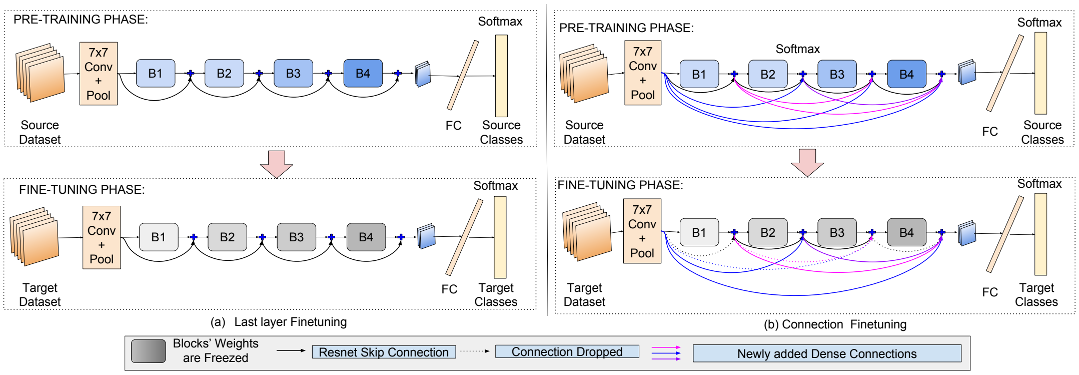

Figure 2: Illustration of process of finetuning of proposed architecture SC-DRCNet vs traditional ResNet on cross datasets.

This architecture, like a traditional DenseNet, assumes all previous connections to be essential to the learning process. However, unlike the traditional DenseNet, instead of concatenating each previous block's output, the new architecture performs summation. The intuition behind this is, if there is a block, whose output has no useful information, in a summation it will simply be ignored. On the other hand, with concatenation, it will be present in the outputs, (and hence inputs) of all succeeding blocks. With a part of the concatenated feature map contributing no information, it may hamper the overall significance of all succeeding connections.

To learn the importance of each connection (add/drop accordingly), we introduce a parameter referred to as "*Connection-Strength*" to encode the value of a connection's contribution [Keshari *et al.*, 2018]. This strength parameter k is taken into consideration with each connection. Now, the output of the n^{th} block is given as:

$$H(x_n) = f(x_n) + \frac{1}{n} \times \sum_{i=1}^{n} k_i \odot x_i \tag{3}$$

where, k_i is the strength associated with i^{th} connection carrying forward input x_i, and $\odot$ is an element-wise multiplication.

3.3 Sparse Connection DRCNet

In order to retain the important connections and remove some of the trivial ones while training the network, l_1 norm has been applied on the strength parameter and optimized along with cross-entropy loss, thus inducing sparsity. This ensures that some strength values are very close to zero, which would essentially be equivalent to those connections being dropped, whereas others which remain would have different values of strength. This is the process of connection-finetuning, wherein the model learns which connections contribute to the final task, but drops those which are unimportant and the same has been depicted in Figure 2. The modified loss function of the model is a combination of traditional Cross Entropy Loss and l_1 norm of the k values, given by the following equation:

$$\mathcal{L}(scores, l) = -scores[l] + log(\sum_{c=1}^{M} exp(scores[c])) + \lambda||K||_{l_1} \tag{4}$$

where, $scores$ variable is the final array of scores for each class obtained after completing a forward pass for given input x, M is the total number of classes, and l is the class label for which the loss is being calculated. λ is the sparsity regularization constant which helps to control the amount of sparsity that is imposed on the vector of all strength parameters ($K = [k_1, k_2, k_3, ...k_n]$). The value of lambda is set in accordance with the significant bits in the CrossEntropy loss of that particular epoch. It starts with a value of 1e-2 and decays at a rate of 0.1.

Similar to ResNet, which has multiple blocks stacked in layers, the proposed architecture also has same number of blocks stacked in multiple Dense Layers. Each Dense Layer has a network of Dense connections, whose strength is learned as shown in Figure 1. This architecture is referred to hereafter as Sparse Connection-DRCNet or SC-DRCNet.

The proposed architecture can also be utilized for finetuning a model pretrained on a source task that needs to be adapted for a target task. Figure 2 is an illustration of conventional finetuning (Figure 2(a)) and the proposed SC-DRCNet finetuning (Figure 2(b)). Usually, in finetuning, only the parameters of the last layer are allowed to be learned via backpropagation. Depending on the availability of data, more layers can be optimized. However, in the proposed SC-DRCNet, strength values of skip-connections are also finetuned. It is worth mentioning that learning the extra strength parameter for each individual connection adds a very small overhead compared to the total number of parameters learned in the training of conventional models. Hence, the total number of learning parameters in the conventional finetuning model and proposed SC-DRCNet model are almost the same. [1]

4 Implementation Details

The proposed architecture is implemented with varying network depths, each a modification of the various ResNet architectures: ResNet-18, 50 and 152. The new architectures are SC-DRCNet-18, 50 and 152, each having the same number of

[1] For example, for model finetuning of ResNet-18 the number of learned parameters are that of the last layer, approximately 512×10 for a 10 class classification problem. For connection-finetuning, an additional 20 strength parameters are required to be learned, thus presenting an extremely small overhead in terms of adding extra learning parameters.

layers as their ResNet counterparts and divided into 2, 4 and 10 Dense Layers respectively. Each Dense Layer consists of 4 or 5 Basic or Bottleneck blocks (identical to ResNet) and has dense skip-connections.

While feature maps are propagated deeper by Dense skip-Connections (as shown by equation 2), the size (length×breadth) of the feature maps output by any current block, is smaller than those brought in by connections from previous blocks. Moreover, as the number of convolutional filters increases while going deeper, the number of feature maps produced are higher than in the previous outputs. However, for summation, all the dimensions must be same. Therefore, the smaller number of feature maps from previous blocks are concatenated with themselves until the number of feature maps are same as that required for summation. To decrease the size of each feature map, average pooling is applied. This made the summation shown in equation 2 possible.

Experiments are performed on one 1080Ti GPU using Pytorch version 0.4.1 [Paszke *et al.*, 2017]. Training is performed for 200 epochs. The initial learning rate is chosen as 0.01, and updated by a factor of 0.1 at epochs 50, 100 and 150. Stochastic Gradient Descent (SGD) optimizer is used with a momentum of 0.9 and weight decay of $5 \times e^{-4}$ along with Cross Entropy Loss. Batch sizes are taken as 64 or 32. For finetuning, weights of all convolutional and batch normalization layers are copied from a Tiny-ImageNet or Imagenet [Deng *et al.*, 2009] pretrained model and frozen during training 2. Only the remaining few parameters are trained.

To reproduce the results of traditional ResNets and DenseNets, the implementation details are kept identical to those in their original papers. Moreover, The testing data has not been used for any parameter tuning. Only the training sets have been utilized to optimize the parameters and to finetune hyperparameters.

5 Experiments and Results

Four benchmarking datasets are utilized to evaluate the proposed DRCNet and “connection-finetuning” in three scenarios: 1) for a database, training SC-DRCNet from scratch along with connection finetuning, 2) pre-training DRCNet on a source database and finetuning skip connections on a target database of the same domain, and 3) finetuning skip-connections on a small target database (i.e. when less training data is available).

5.1 Datasets and Protocols

Table 1: Summarizing the datasets and experimental protocols used for performance evaluation.

Dataset	Total Images	Image Resolution	No. Of Classes	Train Set	Test Set
CIFAR10	60,000	32x32x3	10	50,000	10k
CIFAR100	60,000	32x32x3	100	50,000	10k
SVHN	99,298	32x32x3	10	73,257	26,032
Tiny-Imagenet	110,000	64x64x3	200	100,000	10k

The performance of the proposed algorithm is demonstrated on four datasets: CIFAR10 [LeCun *et al.*, 1998], CIFAR100 [LeCun *et al.*, 1998], Tiny-ImageNet [TinyImageNet, 2018], and SVHN [Netzer *et al.*, 2011]. The experiments and protocols of the respective databases are mentioned in Tables 1 and 2. For ImageNet based experiments, MNIST and SVHN databases are also used for target databases to show transfer learning capability.

Table 2: Summarizing the experimental setup.

Experiment	Source Dataset	Target Dataset
Same Dataset Connection Finetuning	CIFAR10	CIFAR10
	CIFAR100	CIFAR100
	SVHN	SVHN
	Tiny-ImageNet	Tiny-ImageNet
Cross Dataset Connection Finetuning	Tiny-ImageNet	CIFAR10
	Tiny-ImageNet	CIFAR100
	ImageNet	CIFAR10
	ImageNet	CIFAR100
	ImageNet	SVHN
	ImageNet	MNIST
Small Samples Cross Dataset Connection Finetuning	ImageNet	small CIFAR10

As shown in Table 2, to evaluate the proposed method on a small target dataset in adapting from a source task, we selected a small subset as the target dataset for finetuning. The complete procedure is shown in Figure 2. This small sample dataset is assembled from the $50k$ training images of CIFAR-10, by choosing randomly $10, ..., 100, 200, ..., 500.$ samples from each class. We experimented with $100, 200, ..., 1k, 2k, ..., 5k$ as sizes of the subsets.

5.2 Same Dataset Connection-Finetuning

Connection-finetuning has been performed on four databases (CIFAR-10, CIFAR-100, SVHN, Tiny-Imagenet) and three different network depths of the proposed SC-DRCNet architecture (18, 50 and 152). This included three experiments: 1) training a conventional ResNet architecture (Our Baseline model), 2) training a DRCNet architecture, and 3) training a SC-DRCNet architecture from scratch for each dataset. The results are reported in Table 3.

For all four datasets, SC-DRCNet architecture yields improvement in performances. This can be attributed to removal of unnecessary connections which leads to reduced overfitting of the network. However, it is noted that the performance drops with DRCNet architecture which keeps all connections throughout training. This is understandable since unimportant features are forced to remain in the network, thus diminishing it’s ability to generalize well. A similar trend is observed even for deeper networks, i.e. 50 and 152 layers.

On the SVHN dataset, DRCNet-18 is performing slightly better than SC-DRCNet-50. To estimate whether this difference is significant or not, the McNemar Test [McNemar, 1947] is used. Keeping a significance threshold of 0.05, or 5%, we observed that the null hypothesis is accepted, thereby the difference is statistically insignificant. Thus, both architectures perform at par with one another when training from scratch. For other databases, the null hypothesis is rejected. There results demonstrate the improvement in performance of the proposed “connection-finetuning” over traditional skip-connection architectures for learning tasks.

Drawing parallels from novel modifications of ResNet

Table 3: Test accuracy (%) on four benchmarking databases. Results are evaluated on three different depths (18, 50, and 152) of ResNet architecture. Table contains results of DRCNet of their corresponding depth. On the trained DRCNet, spare-connection DRCNet has been trained.

Dataset		C10			C100			SVHN			Tiny-Imagenet		
Model		ResNet	DRCNet	SC-DRCNet	ResNet	DRCNet	SC-DRCNet	ResNet	DRCNet	SC-DRCNet	ResNet	DRCNet	SC-DRCNet
Network Depth	**18**	93.21	91.55	**93.88**	70.79	65.19	**72.5**	91.04	**91.4**	91.314	62.33	55.3	**62.86**
	50	93.22	92.56	**93.60**	70.82	70.7	**73.1**	91.11	87.28	**91.33**	65.13	61.76	**65.61**
	152	93.02	92.8	**93.43**	73.75	72.41	**73.91**	-	-	-	-	-	-

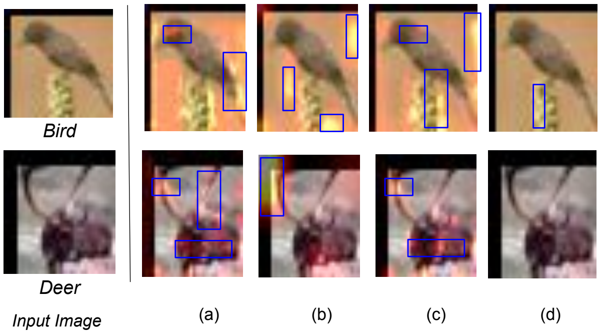

Figure 3: Visualisations of heat maps of an (a) original ResNet connection that was retained, (b) original ResNet connection that was dropped, (c) DRCNet connection that was retained, (d) DRCNet connection that was dropped.

architectures mentioned in the literature (for which reproducible code is available), the proposed SC-DRCNet attains equivalent, if not improved accuracies. Gated-ResNet-32 proposed by Savarese *et al.* [Savarese *et al.*, 2016], documented an accuracy of 93.33% on CIFAR-10, whereas SC-DRCNet-32 achieves an accuracy of 93.45% on CIFAR-10. Similarly, Hettinger *et al.* [Hettinger *et al.*, 2018] reported 72.69% accuracy on CIFAR100, whereas the proposed method yields the 73.1%. Moreover, Huang *et al.* [Huang *et al.*, 2018] reported an accuracy of 93.78% for CIFAR-10 using CondenseNet-50, and the proposed SC-DRCNet-50 also achieves comparable results.

Further analysis of the results showcase interesting observations in terms of the skip connections. In Figure 3, feature maps which have been passed on to successive layers via skip connections are superimposed on the input image to provide a visual representation of the information carried by skip-connections. The areas of high intensity/information appear bright yellow and others are deep red. A feature map with no information (i.e. all zeros) has no heat map signature on the input image. For additional clarity, these areas of significant information are shown bounded by the box. It can be observed that connections highlighting insignificant information or even misinformation are dropped from the network. For example: for the input images of the deer and bird, heat maps in Figure 3(b) mainly carry information about background features and not the animal, which are not integral to the task of identification and hence have been removed. On the other hand, feature maps propagated forward by those connections which are retained, intensify the discriminatory features of the image and thus are preserved as important feature extractors.

Additionally, Figure 4 is the illustration of the frequency distribution of final learnt strength values for the proposed SC-DRCNet while training from scratch on the CIFAR-10 database. It can be noted that about 20-30% of all k values are nullified (very close to zero) and others represent varying degrees of strength/feature importance. This supports our hypothesis that some connections are redundant while others carry important information and should be retained.

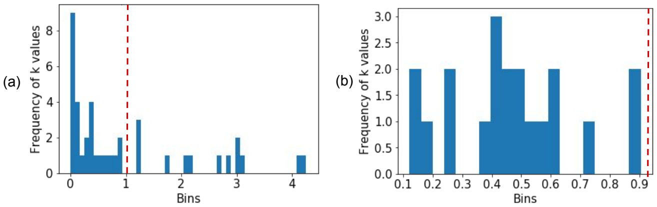

Figure 4: Frequency distribution of the absolute values of the scalar parameter k for networks of depths (a)18, (b)50. The x-axis shows the various bins formed for calculating frequencies and y-axis is the counts of occurrence of k values in these bins. The dotted red line depicts the strength value of skip-connections in a conventional ResNet architecture i.e. 1. All learned strength parameters are different from the default value of 1, thus supporting our hypothesis that skip connections carry information of varying importance.

5.3 Cross Dataset Connection-Finetuning

To evaluate the effectiveness of the proposed algorithm with respect to adaptation in a transfer learning setup, the fol-

Table 4: Test accuracy (%) on CIFAR10 and CIFAR100 datasets. The network is pre-trained on the Tiny-ImageNet database.

Network Depth	**Target Dataset**	**Baseline (ResNet [He *et al.*, 2016])**	**Proposed**
18	CIFAR-10	67.46	**73.64**
	CIFAR-100	43.62	**51.10**
50	CIFAR-10	73.27	**76.81**
	CIFAR-100	48.25	**56.55**

Table 5: Test accuracy (%) on multiple datasets to emphasize performance on transfer learning tasks. The network (depth 50) is pre-trained on the ImageNet dataset [Deng *et al.*, 2009] (source dataset).

Target Dataset $- >$	C10	C100	SVHN	MNIST
Baseline	67.86	59.21	45.16	96.57
Proposed	**77.86**	**67.28**	**63.92**	**98.85**

Table 6: Test Accuracy (%) on CIFAR-10/100 datasets for vanilla DRCNet's of depths 40 and 50 compared with DenseNet-40. The network is pre-trained on the Tiny-ImageNet database.

Target Dataset	**Architecture**		
	DenseNet 40	SC-Dense ResNet-40	SC-Dense ResNet-50
CIFAR-10	70.27	75.57	76.81
CIFAR-100	46.25	55.19	56.55

lowing experiments are performed. A conventional ResNet model is pretrained on the Tiny-Imagenet or ImageNet [Deng *et al.*, 2009] database, following which, only the softmax layer is finetuned for CIFAR-10/100. This is considered as the baseline model. For the proposed model, the same pretrained model is used and the last linear layer is then reoptimized along with learning of all the strength parameters for target data: CIFAR-10/100. These experiments are performed on the following variants: Resnet-18 vs SC-DRCNet-18 and ResNet-50 vs SC-DRCNet-50, and the results are reported in Tables 4 and 5. A boost of 5-10% in test accuracies is observed for adaptations of pre-trained networks, on different targets.

To further showcase the effectiveness with respect to different target databases, network is pretrained on the ImageNet database and finetuned on digit recognition datasets, MNIST and SVHN. As shown in Table 5, the improvement of 2-18% is observed on these databases as well. In the literature, incremental learning through deep adaptation [Rosenfeld and Tsotsos, 2018] has achieved an accuracy of 33.9% on SVHN and 40% on CIFAR-10 dataset. The comparison with these reported results also show that the proposed algorithm yield state-of-the-art performance for domain adaptation task.

The hypothesis behind this appreciable improvement in performance is: without learning the strength/importance of connections for the target samples, the CNN will be forced to work with many features which provide good performance on the source dataset but carry less discriminatory feature information for the target samples. By learning strength values, the network has greater control of which features to give weightage to and which to ignore.

To compare the performance with DenseNet architecture, traditional finetuning is performed on DenseNet-40, to produce results shown in Table 6. Note that the performance of the proposed method with a vanilla architecture of same depth is a great improvement (approximately 6-10%) on DenseNet. The reasoning behind this boost in accuracy remains consistent. Thus, with proposed Connection Finetuning in tandem with traditional finetuning methods, we achieve superior performance, greatly surpassing both - DenseNet and ResNet models.

5.4 Connection Finetuning on Cross Dataset for Small Sample Size

The next set of experiments evaluate the effectiveness of the proposed architecture for small sample databases. For training, the pipeline shown in Figure 2 is followed. The baseline and proposed models are pretrained on ImageNet database [Deng *et al.*, 2009] and then finetuned using a small subset of CIFAR-10 database. Since these subsets are randomly chosen, 5 fold random cross validation is performed and the mean and standard deviations of test accuracies are reported in figure 5. The baseline and proposed models are tested on the complete set of $10k$ images.

Figure 5 shows that the proposed SC-DRCNet-50 yields an improvement of 5 to 20% over standard ResNet-50 for training on small target datasets through varying sizes. It can be seen that standard deviation of test accuracies over the random 5 folds is less for the proposed model (almost 0.25%) than that of the baseline model (almost 0.45%) which demonstrates the robustness against 'variance' of the proposed method of connection finetuning. In other words, better performance using "Connection Finetuning" for learning, even for small sample size, implies that the proposed architecture generalizes easily and adapts better to the target task when training samples are limited. This can be beneficial in a variety of applications where data is difficult to gather or does not exist.

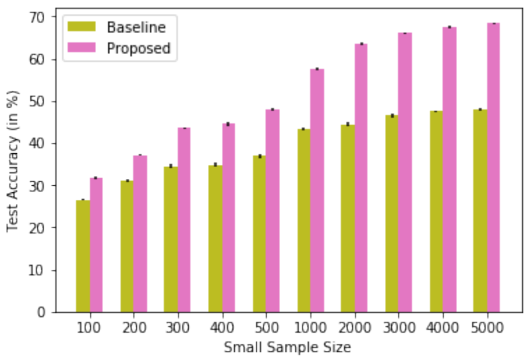

Figure 5: Summarizing the mean and standard deviation of test accuracy over 5 random folds vs database sizes. These results are computed for small dataset with baseline of ResNet-50 and the proposed algorithm as SC-DRCNet-50.

5.5 Ablative Analysis and Observations

Three different kinds of ablative study has been performed: 1) same database, 2) cross database, and 3) random drop of skip-connections.

Same Database Experiments: In order to completely drop or keep a connection, different trials are carried out using various thresholds. For each strength parameter k_i,

$$k_i = \begin{cases} 1, & \text{if } k_i > Threshold. \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Applying equation 5 on all learned k's, their values are frozen (either 0 or 1), thus giving rise to a new architecture each time. The threshold value must be chosen carefully, as a greater threshold may force connections carrying important information to be lost, thus impeding performance, whereas an extremely small threshold value may lead to overfitting. The accuracies for values of thresholds taken as 0.01, 0.1, 0.3 and 0.6, on SC-DRCNet-18 for CIFAR-10 dataset are 93.24%, 93.58%, 93.88% and 93.67% respectively. As compared with a baseline (ResNet-18) performance of 93.22%, it can be observed that a threshold value around 0.3 to 0.6 gives better results.

Secondly, on experimenting with different initializations of Strength parameters, i.e. the k values, in SC-DRCNet, it is observed that initializing with a random double value between 0 and 1, gave a test accuracy of 93.21%, whereas initializing all k's as 1, gave an accuracy of 93.88% on CIFAR-10 with a SC-DRCNet-152. This signifies that learning strength of connections from an unbiased network (i.e. giving each connection equal importance before the learning commences) yields slightly better performance than starting training with random values of connection strength.

Table 7: Test Accuracy (%) on CIFAR10 with different layers (along with Softmax layer) being finetuned. Network (depth 50) is pretrained on the Imagenet database [Deng *et al.*, 2009].

Layers Finetuned	**Baseline**	**Proposed**
Only Softmax layer	67.86	77.86
Last conv layer	73.16	82.95
Last block	80.68	88.03
Last BN Layer	73.04	78.41
All BN layers	87.77	90.00

Cross Dataset Experiments: During finetuning of SC-DRCNet-18 and ResNet-18 on the CIFAR-10 dataset, re-optimizing parameters of different number of layers is performed. The results are reported in Table 7 where, we observed that allowing learning of skip-connection weight parameters (k values) greatly improves the generalizability of the network when not all parameters are learned.

Random Dropping of Skip-Connections: Another set of experiments are performed to signify the importance of learning connection strength or k values by demonstrating that it is not akin to random dropping of connections. To perform the experiments, skip-connections in a DRCNet architecture are dropped at random as follows:

$$k = \begin{cases} 0, & \text{if } prob < threshold \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

where, $prob$ is a random value generated from a uniform distribution and threshold is a hyperparameter which we have finetuned. The above is done for each k value independently and in every training epoch. For threshold values of 0.2, 0.3, 0.4 and 0.5 we achieved 89.54%, 91.65%, 89.39% and 88.9% test accuracy on CIFAR10 with depth 50 architecture. It can be concluded that a threshold value of 0.3 yields best results i.e. randomly dropping 30% connections gives 91.65% accuracy. However, even so it is not at par with our proposed method of intelligently dropping connections by sparsified strength learning which achieves 93.60% (refer Table 3).

6 Conclusion

In this research, we present a novel residual connection finetuning mechanism for the adaptation of deep models to a target task. The proposed approach, termed as “Connection-Finetuning” of Residual Dense Connections, helps in discarding redundant or less important residual connections as well as combining useful feature maps for improved performance. Experimental analysis on multiple databases and architectures have shown that the proposed DRCNet architecture supersedes performance of it’s ResNet counterpart, specifically for the task of transfer learning/cross-dataset finetuning on large as well as small target datasets.

References

[Ahmed *et al.*, 2017] Kaoutar B Ahmed, Lawrence O Hall, Dmitry B Goldgof, Renhao Liu, and Robert A Gatenby. Fine-tuning convolutional deep features for mri based brain tumor classification. In *CAD*, volume 10134, page 101342E. International Society for Optics and Photonics, 2017.

[Casanova *et al.*, 2018] Arantxa Casanova, Guillem Cucurull, Michal Drozdzal, Adriana Romero, and Yoshua Bengio. On the iterative refinement of densely connected representation levels for semantic segmentation. In *CVPRW*, pages 978–987, 2018.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. 2009.

[Guo *et al.*, 2019] Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: transfer learning through adaptive fine-tuning. In *CVPR*, pages 4805–4814, 2019.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Hettinger *et al.*, 2018] Chris Hettinger, Tanner Christensen, Jeffrey Humpherys, and Tyler J Jarvis. Tandem blocks in deep convolutional neural networks. *arXiv preprint arXiv:1806.00145*, 2018.

[Huang *et al.*, 2017] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017.

[Huang *et al.*, 2018] Gao Huang, Shichen Liu, Laurens Van der Maaten, and Kilian Q Weinberger. Condensenet: An efficient densenet using learned group convolutions. In *CVPR*, pages 2752–2761, 2018.

[Keshari *et al.*, 2018] Rohit Keshari, Mayank Vatsa, Richa Singh, and Afzel Noore. Learning structure and strength of cnn filters for small sample size training. In *CVPR*, pages 9349–9358, 2018.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Li and Hoiem, 2018] Zhizhong Li and Derek Hoiem. Learning without forgetting. *TPAMI*, 40(12):2935–2947, 2018.

[Mallya *et al.*, 2018] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV*, pages 67–82, 2018.

[McNemar, 1947] Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *Advances in neural information processing systems-W*, volume 2011, page 5, 2011.

[Paszke *et al.*, 2017] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[Qu *et al.*, 2019] Chen Qu, Feng Ji, Minghui Qiu, Liu Yang, Zhiyu Min, Haiqing Chen, Jun Huang, and W Bruce Croft. Learning to selectively transfer: Reinforced transfer learning for deep text matching. In *ICWSDM*, pages 699–707. ACM, 2019.

[Rebuffi *et al.*, 2017] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pages 506–516, 2017.

[Rebuffi *et al.*, 2018] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *CVPR*, pages 8119–8127, 2018.

[Rosenfeld and Tsotsos, 2018] Amir Rosenfeld and John K Tsotsos. Incremental learning through deep adaptation. *TPAMI*, 2018.

[Savarese *et al.*, 2016] Pedro HP Savarese, Leonardo O Mazza, and Daniel R Figueiredo. Learning identity mappings with residual gates. *arXiv preprint arXiv:1611.01260*, 2016.

[Tajbakhsh *et al.*, 2016] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *TMI*, 35(5):1299–1312, 2016.

[TinyImageNet, 2018] TinyImageNet. Tiny ImageNet tiny imagenet visual recognition challenge. https://tiny-imagenet.herokuapp.com/, 2018. Accessed: 14th-May-2018.

[Yosinski *et al.*, 2014] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NIPS*, pages 3320–3328, 2014.