This WACV 2020 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

ADNet: Adaptively Dense Convolutional Neural Networks

Mingjie Wang[1,2] Hao Cai[1,2] Xin Huang[1,2] Minglun Gong[2*]
[1]Department of Computer Science, Memorial University of Newfoundland, Canada
[2]School of Computer Science, University of Guelph, Canada
mingjiew@mun.ca, hc1864@mun.ca, xhuang@mun.ca, minglun@uoguelph.ca

Abstract

Convolutional neural networks (CNNs) have demonstrated great success in vision tasks. However, most existing architectures still suffer from low feature reuse efficiency. In this paper, we present a layer attention based Adaptively Dense Network (ADNet) by adaptively determining the reuse status of hierarchical preceding features. Specifically, a dense residual aggregation strategy is developed to fuse multi-level internal representations in an effective manner. Furthermore, a novel layer attention mechanism is proposed to explicitly model the interrelationship among layers to automatically adjust the density of the network. It is worth noting that existing ResNets and DenseNets are both special cases of our ADNet. Extensive experiments demonstrate that the proposed architecture consistently and indubitably achieves competitive results in accuracy on benchmark datasets (CIFAR10, CIFAR100, and SVHN), while at the same time remarkably reduces computational costs and memory space. Visualization and analysis on layer-wise attention further provide better understanding on the density of feature reuse in Deep Networks.

1. Introduction

Recently, Convolutional Neural Networks (CNNs) have led a significant success in a wide range of vision tasks [37, 22]. Exploiting highly representative features of networks has given rise to a variety of highly capable deep architectures, starting from AlexNet [20], VGG [28], GoogLeNet [32], Highway Network [30], ResNet [8], DenseNet [14], SENet [11] to Adaptively Connected Network [36] and their variants [22, 44, 34, 35]. The sustaining improvements are mainly attributed to three important factors of CNNs: scale, feature fusion and attention mechanism.

Most of advances in deep learning focus on training very large-scale networks with great depth, width and cardinality. From AlexNet [20] and VGG [28], to ResNet and DenseNet [8, 14], the networks with hundreds of layers have been built. To facilitate the training of deep networks and tackle the issue of feature degradation and gradient vanishing problem, Xavier [4] and He [7] Initializations, Dropout [29], Batch Normalization [17], Stochastic Depth [15] and Group Normalization [38] have been proposed. Additionally, Inception series and Wide Residual Networks (WRN) [44] show that increasing network width helps improve performances since more features can be reused, whereas ResNeXt [39] and Xception [2] demonstrate that the cardinality of the network is also an essential factor.

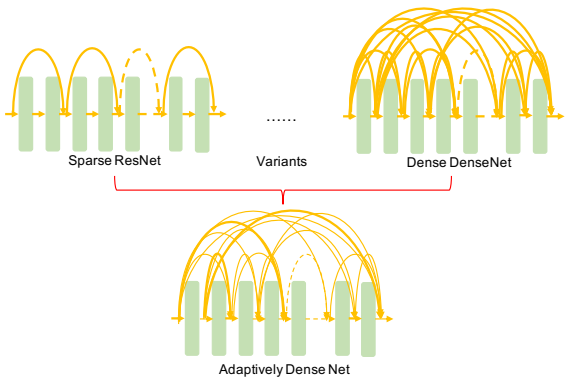

Figure 1. ADNet can be regarded as a generalization of sparse ResNet and dense DenseNet in terms of feature reuse. The density of feature reuse is automatically learned and adaptive to the data.

Apart from the scale of networks, features fusion and attention mechanism also become important aspects in architecture design to efficiently make use of internal representations. Approaches that focus on feature aggregation [8, 14, 43, 23, 46], in particular the ResNet [8] and DenseNet [14], have achieved great success due to the residual and dense connection patterns. ResNet first presents identity mapping to ease the difficulty of training deep network with less parameters, but the representation of each layer in ResNet is only reused by one subsequent layer and lots of layers are redundant [15], significantly reducing the efficiency of learning. To overcome this limitation,

dense feature reuse is proposed in DenseNet [14]. Although DenseNet yields excellent performance, the density of skip connections is extremely large, leading to higher overhead and slower convergence.

Meanwhile, the significance of visual attention has also been studied extensively in recent works like Residual Attention Network [35], SENet [11] and [24, 18, 37, 26]. Attention not only serves to capture the salient objects, locations and channels, but also improves the representation of interests [37]. Nevertheless, existing attention models are generally spatial-wise [18, 35], channel-wise [11], or combination of these two dimensions [37, 1].

This paper presents a novel end-to-end *Adaptively Dense Network*, dubbed ADNet. It provides a generalization among sparse ResNet [8], dense DenseNet [14] and their variants with handcrafted refinements. With the density of feature reuse being automatically learned, ADNet possesses both advantages of effective residual learning and sufficient feature reuse with less redundancy; see Figure 1. Although the learned weights for multi-level reused features are often non-zero and the explicit short connections remain to be nearly dense, the ADNet aims to adaptively search the optimal density (i.e., "connection bandwidths") for reusing features from previous layers [36].

To the best of our knowledge, ADNet is the first attempt to investigate layer interrelationship in structure design and offer a much more compact model. ADNet reuses the preceding feature maps with less redundancy and boosts the learning efficiency by using layer attention mechanism as well as dense residual aggregation for each layer. Our model is built with multiple adaptively dense blocks (ADBs). Each ADB contains a stack of composite layers comprised of dense residual aggregation, layer attention network, ReLU [5], 3×3 convolution and Batch Normalization [17]; see Figure 2. By explicitly modeling the interdependencies among the preceding layers in the layer attention module, skip connections are automatically selected and features are discriminatively reused [35] to emphasize informative layers and suppress less useful ones [11], leading to an adaptive density of feature reuse.

We validate the efficacy of the proposed ADNet on three basic widely-used benchmarks: CIFAR10 [19], CIFAR100 [19] and SVHN [25]. The results show that ADNet outperforms existing architectures (e.g., DenseNet, ResNet and WRN) by a large margin while requiring much fewer parameters. To offer insights, we also visualize and analyze the relationships among multi-level layers in different ADBs and propose a quantitative measure for network density, which indicates the effectiveness of feature reuse.

2. Related Work

Efficient Architectures. The increasing width, depth and cardinality in modern deep networks have shown

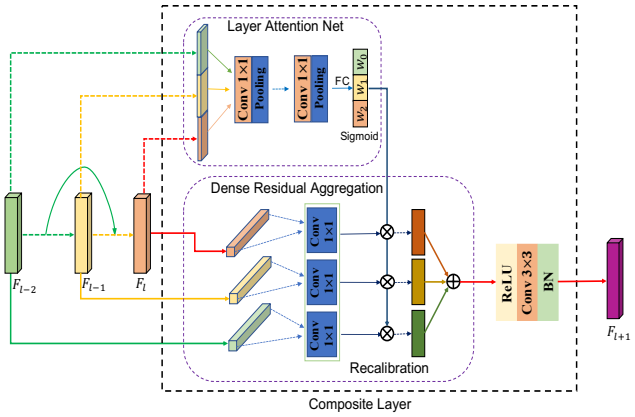

Figure 2. The structure of Composite Layer in ADNet. It is composed of three parts: Layer Attention Network, Dense Residual Aggregation and Convolution operation.

remarkable improvements in performance. VGG [28], GoogLeNets [32], WRN [44] and group convolution-based models [33, 31, 39, 16] illustrate the benefits of increasing depth, width and cardinality. Nevertheless, the extensive redundancy (e.g. filter weights or feature reuse) in large-scale networks usually result in wasting computational costs. To tackle this issue, a range of previous studies have explored efficient end-to-end networks, such as MobileNet [10] and ShuffleNet [45]. Recently, the network compression and pruning have attracted growing attention as efforts to reduce the computational cost and memory requirement of CNN models. Consequently, exploring efficient architectures with superior performance and low overhead is of great importance.

Feature Fusion. To ease the difficulty of training large-scale networks and improve the performance, it has been proved that feature aggregation via skip-connection is effective for a series of vision tasks [23, 43, 22]. Highway Network [30] first provides bypassing paths along with gating units. ResNet [8] achieves record-breaking performance on various tasks by introducing identity mapping. However, its feature reuse rate is low and many layers are redundant. To tackle this issue, the pattern of dense connection is proposed as a way to encourage feature reuse and reduce the model complexity in DenseNet [14]. Despite the stunning results in DenseNets, some feature reuses contribute very little to the final performance [44]. Huang et al. [13, 12] further propose to address this issue by removing superfluous reuses in a hard way, sharing the goal of finding an effective connections strategy for CNNs [36]. Meanwhile, a flexible yet efficient selection mechanism is put forward in DelugeNets [21], which utilizes cross-layer depthwise convolutions. SparseNet [47] introduces a new internal connection pattern which aggregates a greatly sparse set of previous feature maps at any given depth. Unlike these works,

our ADNet facilitates efficient feature reuse by adaptively filtering or recalibrating reused features to determine skip connections.

Attention Mechanism. Attention mechanism has become an emerging trend of research in various tasks [1, 27], with the assumption that human vision does not process an entire input image at once and only focuses on salient parts [3]. The goal of attention is to bias the allocation of available resources towards the most interesting information [24]. Several approaches [35, 40, 41] explore spatial attention in CNNs to select the most informative regions. Hu et al. [11] propose a generic Squeeze-and-Excitation block to rescale channels in convolution layers. In addition, [26, 37, 1] exploit the combination of spatial and channel-wise attentions. Yang et al. [42] develop a hierarchical attention model which has two levels of attention applied at the word and sentence levels. Motivated by this, we propose a useful layer attention module in our ADNet to explicitly model the interdependencies among preceding layers for each convolution layer and adaptively determine which groups of representations should be passed and reused.

3. Adaptively Dense Network

As depicted in Figure 3, ADNet is constructed by a sequence of adaptively dense blocks, denoted as ADBs. Similar to [14, 37, 46], we introduce a transition layer between adjacent blocks, which comprises 1×1 convolution operation and average pooling layer with stride 2. In our experiments, we fix the number of ADBs to 3 and represent the input color image of size $H \times W$ as X_0. For simplicity, we also define the number of filters in each composite layer as growth rate n. As we intend to study the impacts of our new aggregation pattern and layer attention strategy, the bottleneck layer introduced in DenseNet-BC [14] is not adopted.

3.1. Dense Residual Aggregation

ResNets. The conventional residual networks [8] employ the identity mapping to connect the outputs of $(l-1)^{th}$ layer and l^{th} transformation function $F_l(X_{l-1})$:

$$x_l = x_{l-1} + F_l(x_{l-1}), \tag{1}$$

where x_l represents the feature maps extracted by the l^{th} layer. ResNets allow to pass representations and gradients between the earlier and later layers via identity mapping. Although this type of connection effectively eases the difficulty of training deep networks and helps the ResNets achieve huge success, training deep ResNet has a severe issue of diminishing feature reuse [44], which gives rise to superfluous layers.

DenseNets. To cope with the limitation of deep ResNets, DenseNet [14] is proposed and the key idea is to densely connect each layer with all the subsequent layers. Formally, the input of l^{th} layer is denoted as $x_l = F_l([x_1, x_2, ..., x_{l-1}])$, where $[x_1, x_2, ..., x_{l-1}]$ refers to the concatenation of the feature maps generated by $1, 2...l-1$ layers. Although the DenseNet has enhanced feature reuse and reduced the model complexity, this dense connection pattern has several limitations, e.g. 1) the excessive connections not only decrease the parameter-efficiency, but also make the network prone to overfitting [47]; 2) the pattern of dense feature reuse may introduce redundancies when preceding features are not reused by later layers [13].

To tackle these issues and further address the problem of feature degradation, we introduce a novel strategy of features aggregation named as dense residual aggregation, which is motivated by the residual and dense connections. Hence, this fusion approach not only takes advantage of the power of residual learning in ResNets, but also preserves the merits of feature reuse in DenseNets.

Specifically, to match dimensions and separately aggregate local representations of each preceding layer, a set of 1×1 convolutions $\Phi_n^l(x)$ is employed, where n and l are the indexes of the preceding and current layers, respectively. Each 1×1 conv has $k_0 n$ filters, where k_0 is a reduction factor that controls the communication channel width between the input and the current composite layer. The summation is then used to build the layer of dense residual aggregation. Assuming that $[x_1, x_2, ..., x_{l-1}]$ is the input of l^{th} composite layer, the output can be formulated as:

$$x_l = F_l[\Phi_1^l(x_1) + \Phi_2^l(x_2) + ... + \Phi_{l-1}^l(x_{l-1})] \tag{2}$$

We can also transform Equ. (2) to the form of identity mapping:

$$x_l = \left(\Phi_1^{l+1}(x_1) + ... + \Phi_{l-1}^{l+1}(x_{l-1})\right) + F_l(x_{l-1}) \tag{3}$$

where $F_l(x_{l-1})$ represents the residual mapping to be learned. Therefore, our fusion strategy can be interpreted as an ensemble of a series of nested residual units with different widths and lengths, which allows our model to inherit the merits of ResNets – it is easier and faster to optimize residual mapping [8] during the training phase.

3.2. Layer Attention Network

ADNet switches among different densities of feature reuse through an adaptive layer attention module. Following the previous channel-wise and spatial attention mechanisms in SENet [11] and Residual Attention Network [35], our attention module is constructed by alternatively stacking 1×1 conv layer and average pooling, finally followed by global average pooling and sigmoid function. Figure 4 illustrates the layout of the layer attention subnetwork in ADNet.

Dual Dense Connections. Along with the dense skip-connection in dense residual aggregation, we introduce

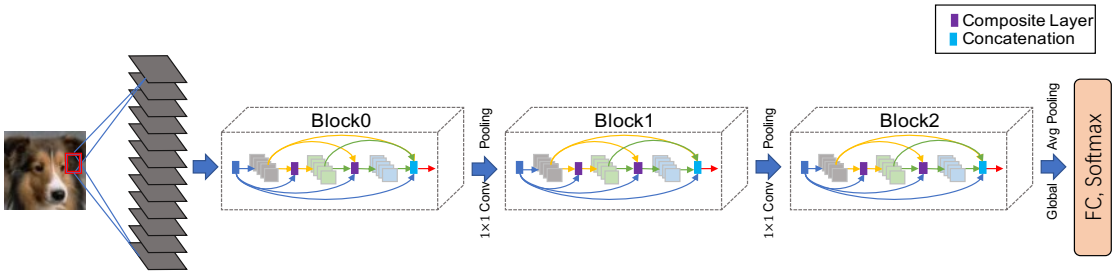

Figure 3. The overall schema for ADNet architecture.

the additional dense skip-connection for layer attention. The input of attention model is formulated as $I_l = [x_1, x_2, ..., x_{l-1}]$, where x_l refers to l^{th} preceding feature and $[,]$ denotes concatenation operation. The connection pattern in our attention module differs greatly from existing attention algorithms. The goal of this design is to effectively mitigate the problems of features degradation and gradient vanishing, to make each attention module work independently, and to break co-adaptation of layers. Intuitively, stacking attention layer always leads to performance drop as dot product with mask metrics from 0 to 1 repeatedly will seriously degrade the values of features and gradients [35] in deep layers. This connection pattern helps avoid this problem resulted from attention mechanism by enabling the forward features and backward gradients to be directly propagated among different layers.

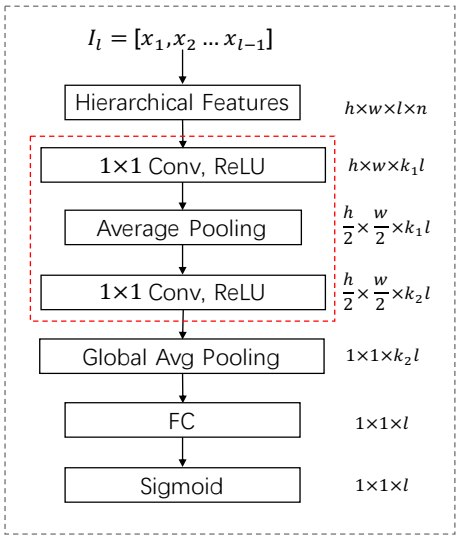

Figure 4. Layer Attention SubNetwork.

Layer Attention. Although dense residual aggregation is capable of implicitly extracting the layers interdependencies via 1×1 convolution and summation operations, these relationships are entangled with the spatial and channel correlation. We propose a transformation to explicitly model layers interrelationships and determine features reused by the current composite layer. Rather than directly applying global average pooling to capture global information in SENet [11], we set up three layers before global average pooling to fuse cross-channel features and extract spatial information; see Figure 4. This design aims to facilitate more discriminative representations for layer relationship modelling.

As presented in Figure 4, from the input I_l, the first 1×1 operation $\Phi_1(x)$ is applied to aggregate the cross-channel information and reduce the dimension to $k_1 l$, where l indexes the current layer. In order to increase the non-linearity of representations, activation function ReLU is employed and represented as $\delta_1(x)$. Average pooling function $P_{2\times2}(x)$ is then performed to increase the receptive field of features, followed by two consecutive operations: the secondary 1×1 conv $\Phi_2(x)$ and ReLU $\delta_2(x)$. Finally, a global average pooling $P_{global}(x)$ encodes the global structure information and its output is fed into a fully-connected layer $f(x)$ with $k_2 l$ channels. Then the sigmoid function normalizes the output range to [0,1]. We refer to the hyper-parameters k_1 and k_2 as the reduction factors in attention subnetwork. The weight vector is generated to filter multi-branch reused representations $\Phi_i(x_i)$, $i \in (1, 2, ..., l-1)$ in Dense Residual Aggregation. Consequently, the output of l^{th} dense residual aggregation is refomulated as:

$$s_l = w^l X_l = \sum_{i=1}^{l-1} w_i^l \times \Phi_i^l(x_i), s_l \in \mathbf{R}^{H^l \times W^l \times C^l}, \quad (4)$$

where $X_l = (\Phi_1^l(x_1), \Phi_2^l(x_2), ..., \Phi_{l-1}^l(x_{l-1}))$ and $\times$ denotes layer-wise multiplication. Thus, we can modify output of x_l composite layer Equ. 2 as follows:

$$x_l = F_l(w^l X_l) = F_l[w_1^l \Phi_1^l(x_1) + w_2^l \Phi_2^l(x_2) + ... + w_{l-1}^l \Phi_{l-1}^l(x_{l-1})], \quad (5)$$

where weight vector w^l represents the interdependencies of

preceding layers and ranges from 0 to 1 to softly control the status of feature reuse. The layer attention helps generate a new density pattern during each mini-batch and finally an adaptively dense network is automatically learned.

4. Experiments

We evaluate ADNet on three standard benchmarks (CIFAR10, CIFAR100 and SVHN) for image classification task. The results suggest that our ADNet is more efficient and compact than several state-of-the-art architectures. In particular, we reproduce the results of DenseNets and use them as our comparison baseline, because DenseNet outperforms WRN, ResNet, and ResNet's variants in both accuracy and parameters effectiveness.

Datasets. The CIFAR10 and CIFAR100 [19] datasets consist of 60,000 (50,000 for training and 10,000 for testing) 32×32 color images with 10 and 100 classes respectively. SVHN dataset is another widely-used one and consists of 32×32 color digits images, from 0 to 9, with 73,257 training samples, 26,032 for testing and 531,131 additional training images accordingly. We adopt a standard data augmentation of random cropping with 4-pixel padding and horizontal flipping for CIFAR and normalize them by using mean and standard deviation values, whereas for SVHN images, the pixels values are divided by 255.

Implementation. In our experiments, an initial 3×3 convolution layer is set up before the first block to gain the basic image transformations. There are three reduction factors in our model: k_0 is set to 4 in multi-branch aggregation and $k_1 = 8, k_2 = 4$ for layer attention module. Our network is trained in an end-to-end manner using SGD optimizer with momentum 0.9 and a mini-batch size of 64. We train 300 epochs for CIFAR and 40 for SVHN from scratch using the He initialization [7]. Initial learning rate is set to 0.1 and progressively divided by 10 at epochs 150, 225 for CIFAR and epochs 20, 30 for SVHN. Batch Normalization [17] and dropout [29] with drop rate of 0.2 are applied in ADNet. We report test error from the epoch with the lowest validation error.

4.1. Classification Results

First we compare ADNet with different architectures and parameter settings. The test errors on three datasets are listed in Table 1. The results show that, compared with DenseNets, our model obtains performance gains under various hyper-parameter settings. ADNet produces error rates 3.84% on C10+ (n=40, L=28) and 20.20% on C100+ (n=40, L=36). They are lower than the counterparts achieved by DenseNets (4.50% on C10+ and 22.05% on C100+), while reducing the amounts of parameters by 12.96% and 15.85% respectively. Figure 5 shows the training (top) and testing (bottom) curves of ADNet and DenseNet on C10+. Figure 6 further compares the two approaches in terms of both accuracy and computational cost. It shows that ADNet achieves better accuracy with only requiring $1/3$ or fewer number of FLOPs than DenseNet.

Furthermore, compared with Stochastic Depth-1002 (10.2M), ResNet-1001 (10.2M), FractalNet (38.6M), and WRN-28 (36.5M), our ADNet **(6.9M)** with $n = 40, L = 36$ yields a consistent reduction in test errors on all CIFAR and SVHN datasets. Impressively, we observe that the improvements of our ADNet are particularly pronounced on the original C10 and C100 datasets without any data augmentations. Under the setting of $n = 40$ and $L = 36$, our accuracy improvement over those produced by ResNet-1001, Fractal Net, and DenseNet ($n = 24$, $L = 100$) is 51.4%, 30%, and 12%, respectively, on C10 and 38.60%, 27.23%, and 12.38%, respectively on C100. Additionally, our ADNet ($n = 40, L = 36$) achieves the best result of 1.54% on SVHN. Although it is equal to the error reported by WRN, our model is more light-weight (6.9M vs. 11.0M), resulting in lower computational overhead.

With increasing depths and growth rates, error drops from 5.34%, over 4.40% to 3.84% on C10+, from 24.10%, over 22.59% to 20.20% on C100+, and from 1.68%, over 1.59% to 1.54% on SVHN. This indicates that the proposed ADNet is not prone to overfitting except for the case that the increasing the depth 28 to 36 leads to a modest increase in test error on C10+ from 3.84% to 3.96%.

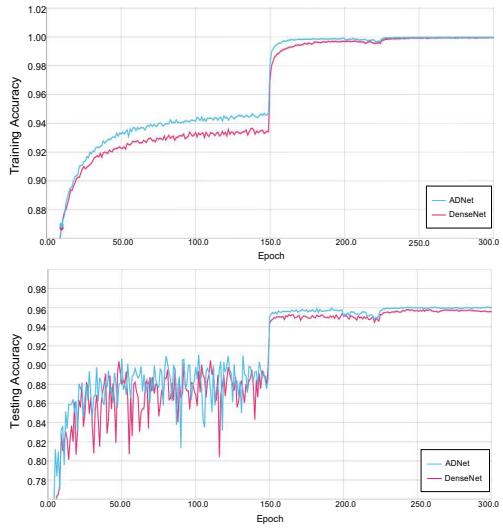

Figure 5. Training and testing curves of ADNet and DenseNet with the configuration of ($n = 40$, $L = 28$).

Model Complexity. ADNet provides a better trade-off between the complexity and performance, leading to reductions in both error rates and the amount of parameters. Figure 6 shows that our model is highly cost-effective. In order to compare the computational costs of ADNet and DenseNet, the numbers of parameters introduced by each

Model	Depth	Params	C10	C10+	C100	C100+	SVHN
Fractal Network [22]	21	38.6M	7.33	4.60	28.20	23.73	1.87
ResNet [8]	110	1.7M	13.63	6.41	44.74	27.22	2.01
Stochastic Depth [15]	1002	10.2M	-	4.91	-	-	-
ResNet(pre-act.) [9]	1001	10.2M	10.56	4.62	33.47	22.71	-
WRN-16 [44]	16	11.0M	-	4.81	-	22.07	1.54
WRN-28 [44]	28	36.5M	-	4.17	-	20.50	-
ResNeXt [39]	29	0.8M	-	6.74	-	26.48	-
SparseNet($n = 12$) [47]	40	0.8M	-	5.13	-	24.65	-
SparseNet($n = 24$) [47]	100	2.5M	-	4.64	-	22.41	-
SparseNet($n = 36$) [47]	100	5.7M	-	4.34	-	20.50	-
DenseNet($n = 12$) [14]	40	1.0M	7.00	5.24	27.55	24.42	1.79
DenseNet($n = 12$) [14]	100	7.0M	5.77	4.10	23.79	**20.20**	1.67
DenseNet($n = 12$) [14]	28	0.5M	7.36*	6.09*	29.17*	27.67*	1.82*
Our ADNet($n = 12$)	28	**0.6M**	**5.99**	**5.34**	**25.57**	**24.10**	**1.68**
DenseNet($n = 24$) [14]	28	2.0M	6.58*	4.83*	26.16*	24.56*	1.79*
Our ADNet($n = 24$)	28	**1.9M**	**5.23**	**4.40**	**23.20**	**22.59**	**1.59**
DenseNet($n = 40$) [14]	28	5.4M	5.99*	4.50*	25.78*	22.20*	1.71*
Our ADNet($n = 40$)	28	**4.7M**	**5.20**	**3.84**	**21.86**	**20.51**	**1.59**
DenseNet($n = 40$) [14]	36	8.2M	6.15*	4.30*	24.88*	22.05*	1.63*
Our ADNet($n = 40$)	36	**6.9M**	**5.13**	**3.96**	**20.52**	**20.20**	**1.54**

Table 1. Test error (%) on CIFAR and SVHN datasets. Contents in boldface indicate the best results under similar parameter settings. "C10+" and "C100+" indicate the datasets with standard data augmentations used in [14] and n denotes the growth rate [14]. Error rates of existing approaches are reported by the respective papers, except the data with " $*$ " are measured using our reimplementation. ADNet indubitably outperforms DenseNet, ResNet, and its variants under compatible parameters, especially on the datasets without augmentation.

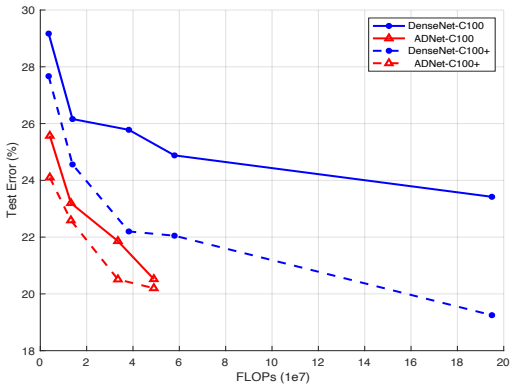

Figure 6. Comparison of the ADNets and DenseNets in terms of accuracy v.s. computational costs on both C100 and C100+ datasets. The FLOPs numbers are reported by TensorFlow.

Adaptively Dense Block (N_{ADB}) and conventional Dense Block (N_{DB}) can be formulated as bellow:

$$N_{ADB}(L) = 9nL + (k_0 n^2 + k_1 n + k_1 k_2 + k_2)\frac{L(L+1)}{2}$$
$$N_{DB}(L) = 9n^2[L + (L-1) + ... + 1] = \frac{9n^2 L(L+1)}{2}, \quad (6)$$

where n and L denote the growth rate and block depth, respectively. k_0, k_1, k_2 are reduction factors in ADNet. Given $n = 40, k_0 = 4, k_1 = 8$ and $k_2 = 4$, we obtain the final parameter numbers as follows:

$$N_{ADB}(L) = 3378L^2 + 3738L$$
$$N_{DB}(L) = 7200L^2 + 7200L, \quad (7)$$

according to Equ. (7), analytically, the number of parameters in ADNet is less than half of those in DenseNet under arbitrary values of depth L.

We also compare the computational complexity of our ADNet with several existing popular methods using the scheme introduced in SparseNets [47], see Table 2. The computational complexity is analyzed from three important aspects: Parameters, Shortest Gradient Path (SGP) and Aggregated Features (AF). Although our ADNet has smaller computational burden than that of original DenseNets, its parameter scale ($O(\frac{1}{2}L^2)$) is higher than that of ResNets and SparseNets ($O(L)$) as the patterns of feature reuse in these two models are sparse. Note that our ADNet is consistent with ResNets and DenseNets in terms of SGP and AF, which shows that our designed layer attention module does not degrade back propagation of gradients and feature-reuse capability.

Ablation Study. In order to demonstrate the impact of proposed attention mechanism, we also evaluate the performances of our root network with and without layer attention modules. Table 3 provides the comparison results on the

Models	Parameters	SGP	AF
ResNets	$O(L)$	$O(1)$	$O(l)$
SparseNets(sum)	$O(L)$	$O(log(L))$	$O(log(l))$
DenseNets	$O(L^2)$	$O(1)$	$O(l)$
ADNet	$O(\frac{1}{2}L^2)$	$O(1)$	$O(l)$

Table 2. Comparison of computational complexity in terms of Parameters, Shortest Gradient Path (SGP) and Aggregated Features (AF)). L denotes the depth of networks and the Aggregated Features (AF) represents the reused features gathered by current l^{th} layer.

Dataset	Growth rate	DenseNet	w.o. LA	w. LA
C10	n=12	7.36	6.30	5.99
	n=24	6.58	5.52	5.23
	n=40	5.99	5.42	5.20
C100	n=12	29.17	26.12	25.57
	n=24	26.16	23.89	23.20
	n=40	25.78	22.01	21.86
SVHN	n=12	1.82	1.75	1.68
	n=24	1.79	1.65	1.59
	n=40	1.71	1.64	1.59

Table 3. Test Error (%) of our 28-layer ADNet with and without layer attention (LA) on C10, C100, and SVHN.

datasets C10, C100, and SVHN. It shows that the layer-wise attention subnetwork leads to performance gains of ADNet. Therefore, the appropriate density of feature reuse or skip connections contributes to boost the performance of CNNs while the pattern of fully dense connections is prone to computational inefficiency and overfitting.

4.2. Model Analysis

Reduction Factors. In ADNet, we introduce three factors, k_0, k_1 and k_2. Specifically, k_0 plays two roles in multi-branch aggregation: reducing the width of input layer and scaling the dimension of internal layer, whereas k_1 and k_2 are used in layer attention module to reduce the number of filters. For simplicity, we uniformly refer to them as reduction factors which control the representative capacity and the model complexity. The proper setting of factors not only provides trade-offs between performance and cost, but also prevents overfitting. After various experiments with different factors, we fix the values: $k_0 = 4$, $k_1 = 8$ and $k_2 = 4$.

Dense Residual Aggregation. To fuse and align hierarchical features for reusing, 1×1 conv layers are utilized to process all branches of reused features individually. Input layer with very large width of each block is narrowed to remove the superfluous channels, while the critical 3×3 conv layers are widened to increase the flexibility of fusion. Summation operation is employed to aggregate hierarchical feature maps, which increases our network's invariance against scale/shift and is the primary reason that our network has better performance gains on datasets without augmentation. Our finding contradicts the previous belief that summation may impede the information flow [14]. Additionally, the summation helps the ADNet possess the advantages of residual learning, leading to a fast convergence speed.

Layer Attention Behavior. While the layer attention network has been empirically illustrated to improve the performance, it is also necessary to understand how the attention mechanism operates practically. We set up additional 1×1 convolution and average pooling layers at top of the module to further extract spatial features while SENet directly utilizes global average pooling to disentangle the impacts of spatial information. Empirical observation demonstrates that the local spatial features are crucial for modeling layer dependencies. The goal of two 1×1 convolutions is to capture the cross-channel information under two scales. Due to the sufficient spatial and cross-channel information, the attention mechanism can learn weight vectors more precisely for each composite layer. The set of learned layer weights in ADNet determines the status of connections and helps generate an adaptive density of feature reuse.

Attention Visualization. To further provide a clear picture on the distributions of layer weighs produced by layer attention and practically understand the status of feature reuse in ADNet, we select four real color images from four classes (aircraft, bird, dog and truck) as inputs of the ADNet and obtain the corresponding layer weights at three blocks. The frameworks ($n = 40$, $L = 28$) trained on C10+ and C100+ are used to test these examples respectively, and consequently the generated distributions are shown in Figure 7. On the simpler task C10+, we observe that the weights obtained for different classes are almost identical in lower blocks but show differences in higher blocks. This suggests that low-level features, such as edge and corner, are likely to be shared by all classes, whereas higher-level features are more class-specific. This observation is similar to the one previously made in SENet [11].

The trend differs on the more complicated C100+ dataset that has lower classification accuracy. Similar distributions only occur in a few top layers in each block. This shows that the basic representations are not sufficiently extracted, which consequently leads to a bad accuracy. Besides, the variances of layers weights are much higher than counterparts on C10+. Possible explanations are: 1) the network is too shallow to represent adequate low-level features; 2) The trained model is not compact enough and there are redundant reuses in shallow blocks which are prone to noise. This seems to explain why the same network architecture has different behaviors on datasets with various complexities.

Feature Reuse Density. We also carry out an experi-

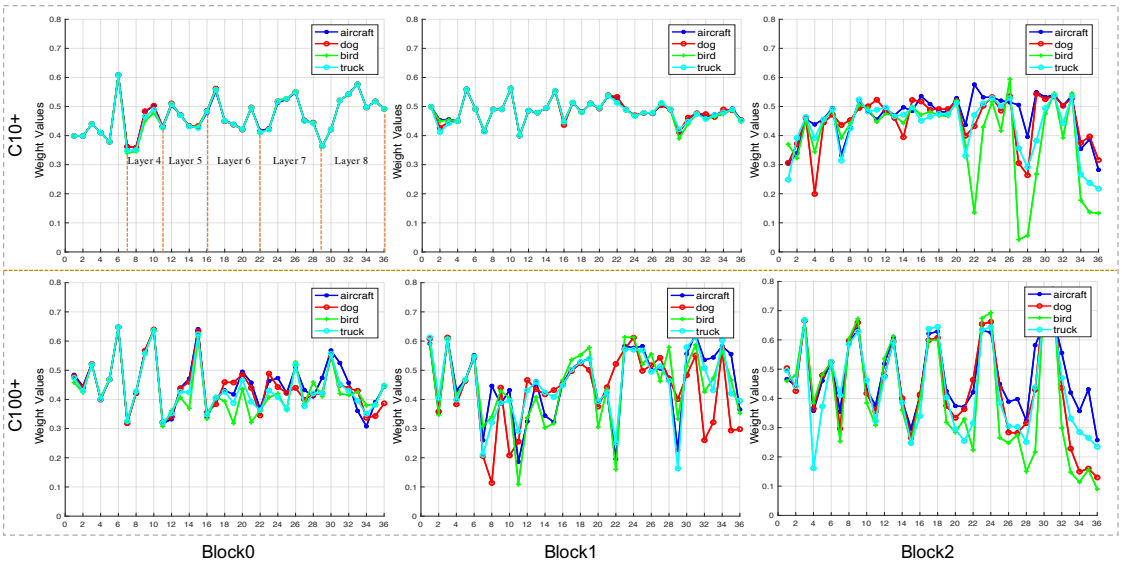

Figure 7. Attention weight distributions produced by layer attention modules of ADNet in different blocks. Vertical axis shows the learned weight values and horizontal axis denotes all preceding layers. On simpler dataset C10+, the weights values obtained for different classes are similar, especially in lower blocks. This suggests that basic features shared by different classes are extracted. On more difficult C100+ dataset, the variations on learned weights are much higher.

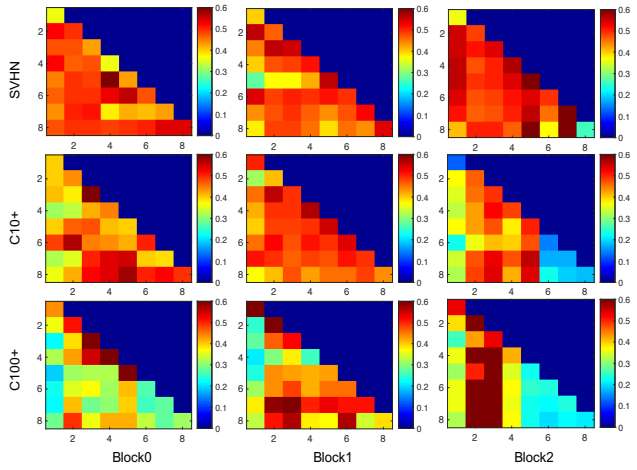

Figure 8. The average layer weights in ADNet based on test images in SVHN, C10+, and C100+ datasets, respectively. On simpler dataset (SVHN), the average weights have relatively high values (red colors) while on more complex C100+, the weights show strong variations, indicating that ADNet is capable of automatically determining the status of feature reuse. Based on datasets with different complexities, our method enables to generate corresponding adaptively dense connections.

ment to evaluate the densities of feature reuse in ADNet trained on different datasets. Three groups of test samples in C10 , C100 (10,000 images), and SVHN (26,032 images) are individually fed into the corresponding trained ADNets ($n = 40$, $L = 28$) and then the mean values of layer weights for all datasets are computed. To better understand the adaptive densities on different datasets, we take advantage of heatmaps to show the layers weights during each block, see Figure 8. Interestingly, we observe that, on simpler datasets (C10 and SVHN), the average layer weights are consistently higher than those on complex C100, which indicates that the densities of feature reuse in our ADNet are automatically learned and adaptive to different datasets instead of fixed or handcrafted reuse pattern.

5. Conclusion

In this paper, we propose a novel adaptively dense neural network, *ADNet*, to improve the representative power and feature reuse efficiency. This is achieved through adaptively attenuating the hierarchical representations from preceding layers and dense residual aggregation after layer-wise recalibration. Extensive experiments illustrate the effectiveness of ADNet on multiple datasets. More importantly, the novel layer attention mode provides means for modeling and visualizing the interdependencies of multi-level layers, which may contribute to other tasks reusing multi-scale features. Additionally, the density study based on layer attention may be used to understand the behaviour of deep networks and design targeted network pruning [6]. In the future, it is worth to explore the methods of optimizing network architectures from the viewpoint of *Feature Reuse Density*. For example, the reuse density can be added into the loss function to favor networks with higher reuse redundancy.

References

[1] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5659–5667, 2017.

[2] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[3] M. Corbetta and G. L. Shulman. Control of goal-directed and stimulus-driven attention in the brain. *Nature reviews neuroscience*, 3(3):201, 2002.

[4] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[5] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.

[6] A. N. Gomez, I. Zhang, K. Swersky, Y. Gal, and G. E. Hinton. Targeted dropout. 2018.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[11] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[12] G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, and K. Q. Weinberger. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*, 2017.

[13] G. Huang, S. Liu, L. Van der Maaten, and K. Q. Weinberger. Condensenet: An efficient densenet using learned group convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2752–2761, 2018.

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[15] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016.

[16] Y. Ioannou, D. Robertson, R. Cipolla, and A. Criminisi. Deep roots: Improving cnn efficiency with hierarchical filter groups. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1231–1240, 2017.

[17] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[18] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[19] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[21] J. Kuen, X. Kong, G. Wang, and Y.-P. Tan. Delugenets: deep networks with efficient and flexible cross-layer information inflows. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 958–966, 2017.

[22] G. Larsson, M. Maire, and G. Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*, 2016.

[23] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[24] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.

[25] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[26] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon. BAM: bottleneck attention module. *arXiv preprint arXiv:1807.06514*, 2018.

[27] A. Show. Tell: Neural image caption generation with visual attention. *Kelvin Xu et. al.. arXiv Pre-Print*, 23, 2015.

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

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[30] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[31] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In

Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2818–2826, 2016.

[34] S. Targ, D. Almeida, and K. Lyman. Resnet in Resnet: Generalizing Residual Architectures. *arXiv preprint arXiv:1603.08029*, pages 1–7, 2016.

[35] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2017.

[36] G. Wang, K. Wang, and L. Lin. Adaptively Connected Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[37] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[38] Y. Wu and K. He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[39] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.

[40] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*, pages 451–466. Springer, 2016.

[41] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.

[42] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

[43] F. Yu, D. Wang, E. Shelhamer, and T. Darrell. Deep layer aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2018.

[44] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[45] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.

[46] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2472–2481, 2018.

[47] L. Zhu, R. Deng, M. Maire, Z. Deng, G. Mori, and P. Tan. Sparsely Aggregated Convolutional Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 186–201, 2018.