On the sparsity and diversity of densely connected networks

***Abstract*—Deep Neural Network community, has been working for the betterment of CNN architecture for a long time, solving innumerable computer vision tasks has become effortless for this. Densely Connected Networks [1] revolutionized the field by introducing the concept of dense connections for better classification performance. However, not all the connections in the model provide important information. In this work, we perform a rigorous analysis of the affect of reducing connections in DenseNet. We propose a novel connectivity pattern to maintain low and high complexity features while keeping the number of connections as low as possible. We also propose weighted skip connections that introduce more diversity to aid in the performance of the model**

***Index Terms*—Deep Neural Network, CNN architecture, skip connections**

I. Introduction

DEEP neural networks have proven to be highly efficient mathematical models in numerous prediction and classification tasks [2]. The introduction of connections [3] that connect distinct hidden layers far away from each other has immensely aided in making the networks deeper as compared to the early networks ([4], [5]), with good convergence properties. This has helped in reducing the top-5 classification error rate from 15.3% to 3.57% on the ImageNet dataset [6]. Densely connected networks [1] leveraged the skip connections to connect the output of a hidden layer to multiple subsequent layers. Thereby, utilizing both the low complexity and high complexity features in improving the classification accuracy. Although ingenious, the DenseNet suffers from the problem of redundant connections between layers as not all connections are important for classification [7]. The large number of skip connections also imply a burden on the computational resources especially of the lightweight devices with limited computational capability. Reducing the size (parameters, layers, filters or nodes) of a neural network to can lead to significant savings in computational cost and time. It can also lead to less overfittting as most of the deep neural architectures are too overparameterized. To this end, several techniques have been proposed to reduce the size of the network.

In this paper, we take a two fold approach in analysing the sparsity and diversity properties of densely connected networks. Hence, we specifically design experiments to alter the connectivity patterns by removing different skip connections within the dense blocks. We also perform a rigorous analysis of the effect of reducing skip connections on the classification accuracy of DenseNet. We also propose and analyse the effect of a novel formula for decreasing redundant connections in DenseNet without an immense degradation of classification accuracy. We also provide an analysis of FLOP savings and accuracy achieved in the reduced networks. We also propose a novel approach to gauge the importance of skip connections by introducing weights on the skip connections, which we term as weighted skip connections. The weights introduced are learned through the backpropagation rule and thus provide a reasonably robust measure of connection importance. These weights also aid in increasing feature diversity which helps the model to achieve more robust classification results on benchmark datasets.

The results show that by removing the skip connections the number of trainable parameters get reduced by a factor of 20.9%,37.9%,47.4%whereas there is only 0.02%,1.14%,1.4% respectively drop in classification accuracy. The weighted skip connections increase only a few hundred trainable parameters whereas improve the classification accuracy by 0.62%.

II. Related work

Deep neural network architectures. AlexNet [4] revolutionized the field of deep learning by winning the 2012 ILSVRC challenge by using an 8 layer network that was trained on multiple GPUs. The usage of group convolution made the training possible on multiple GPUs. AlexNet suffered from the problem of overfitting as it consisted of about 60 million parameters. To deal with this, data augmentation and dropout were also performed. The Visual Geometry Group's proposed architecture, VGG [5] went further ahead in increasing the depth of the network upto 16-19 layers while keeping the number of parameters not very high. This was achieved through the use of consecutive small $3 * 3$ filters. Another disadvantage of increased network depth is the problem of vanishing gradients wherein the gradient coming to the earlier layers has a very small value. This in turn stalls the updation of network weights. Residual networks [3] dealt with this problem by introducing residual connections that learned a residual mapping instead of unknown functions and it aided in the flow of gradient to earlier layers. ReseNet suffered from the usage of addition operator which may impede the gradient flow [7]. DenseNet [1] proposed by Huang et. al. removes the drawback of ResNet by using the feature concatenation

Identify applicable funding agency here. If none, delete this.

operator. DenseNet also introduced identity skip connections to connect a layer to all of its subsequent layers within a dense block. The architecture of DenseNet alleviated the vanishing gradient problem, increased feature reuse with reduced number of parameters.

Pruning and Sparsifying deep networks. Unimportant connections in deep networks can introduce redundant information that can cause overfitting. To deal with this, Han et. al. proposed a novel method [8] that was inspired by the working of the mammalian brain which learnt important connections along with weights. Han et. al. [9] proposed a three stage pipeline to reduce the computational as well as memory requirements of deep neural networks. In order to utilize the diverse neurons and reduce the redundant neurons, Mariet et. al. [10] proposed to use the Determinantal Point Process to identify diverse neurons. Liu et. al. [11] proposed a novel method of introducing sparsity into the deep network through the use of sparse decomposition. A fine tuning step was also introduced wherein recognition loss is minimized as a result of maximizing sparsity. Kim et. al. [12] proposed a novel method for network compression by applying rank selection with variational Bayes matrix factorization followed by Tucker decomposition on kernel tensor and fine-tuning. Rui et. al. [13] proposed a multi stage weighted feature skip connections for head detection. Han et. al. [9] proposed a three stage pipeline that performed network pruning, weight quantization and Huffman coding to compress the neural network size without trading-off with accuracy. The unimportant connections are pruned which reduces the chances of overfitting. Instead of pruning or sparsifying weights of a network, Li et. al. [14] proposed a strategy to directly prune convolution filters, thereby significantly reducing the size of the network.

Wen et. al. [15] proposed a novel method utilizing group LASSO regularizer to regularize filter, channels, filter shape and depth to introduce sparsity into the model. Han et. al. [16] focused on reducing the computation of fully connected (FC) layers by leveraging sparsity in activations and weights. Weight sharing and quantization were also utilized to propose an efficient inference engine for compressed networks. Srinivas et. al. [17] proposed a novel method to prune and learn weights simultaneously. The learning based approach to achieve model sparsity aids in obtaining the optimal level of sparsity. He et. al. [18] proposed a two step approach in reducing the size of the network wherein in the first step, redundant channels are removed based on the LASSO regression technique. In the next step, the output was reconstructed using the remaining channels and the least squares model. Liu et. al. [19] proposed a network slimming technique that utilized the scaling factors from batch normalization layers to regularize the channels to induce sparsity in the network. The sparse channels are then pruned, thus, reducing the network size. Anwar et. al. [20] leverage the intra-kernel sparsity to reduce the complexity of convolution neural networks. Howard et. al. [21] proposed the use of depth-wise separable convolutions that significantly reduce the computations. Two hyperparameters were used to trade-off between accuracy and latency of the model. Huang et. al. [7] combined dense connectivity with learned group convolutions to learn unnecessary connections between layers. Xiao et. al. [22] proposed a weighted connection scheme and skip connections on a URes-Net like model for retinal vessel segmentation wherein the weights aided in pruning unnecessary connections.

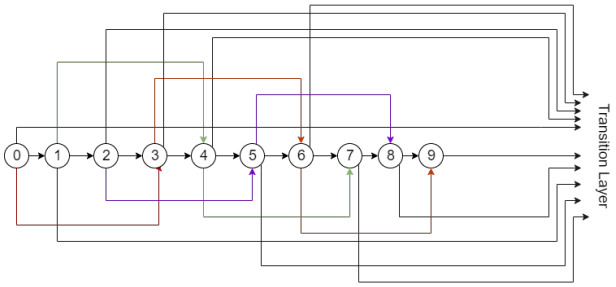


Fig. 1. a nine layer dense block

III. Materials and methods

DenseNet is made by multiple dense blocks, each dense block contains several layers. The output of all preceding layers is received and concatenated by each layer in a dense block. To make the concatenation process smoother they maintained the same feature size within a dense block. A single layer in a dense block can produce K feature map which is known as growth factor. Between two consecutive dense blocks, a transition layer is introduced to downsample feature size. A transition layer is made with a $1*1$ Conv layer and $2*2$ average pooling layer, the last one is preceded by the first one. the model takes care of gradient flow to earlier layers. DenseNet can also be distinguished from other CNN architectures from its property of reusing the features extracted in earlier layers.

Root-n connectivity is the proposed connectivity pattern in this paper.We have used custom function sqrt(n)(n represents the number of layers in the block) to reduce the number of skip connections. For example, if a dense block has 9 layers in fig. III in it then sqrt(9) is 3 which implies that the input layer(0 in the figure) is connected with (0+3=3) 3rd layer(3 in the figure). Likewise, the 3rd layer is connected with 6th layer and 6th layer is connected with the 9th layer. The 1st layer(1 in fig) is connected with the 4th layer by the skip connection,4th is connected with 7th layer. The 7th layer has no skip connection getting out of it as the block does not has the 10th layer. Proceeding like this we are achieving our target of, reducing number of skip connections, maintaining strong gradient flow and feature diversity at the time of classification. We have used black lines in the fig show the skip connection of features of each layer with the transition layer and output of each layer with the subsequent layer. A single color is used to indicate feature flow for each layer in the block.

IV. Experiments

A. Datasets

CIFAR10. The CIFAR10 dataset [23] consists of 32x32 pixel coloured images drawn from 10 classes. The training and testing set contain 50,000 and 10,000 images respectively and we hold 5000 images from training set as validation set. We use standard data augmentation scheme of mirroring and shifting that is widely used [3], [24]–[30] for this dataset.For preprocessing, the data is normalised using the channel means and standard deviations. We select the model with the lowest validation error during training and report the test error.

SVHN. The Street View House Numbers (SVHN) dataset [31] contains 32×32 pixel coloured digit images. The training set and test set contain contain 73,257 and 26,032 images respectively. There are an additional 531,131 images of less difficult samples for additional learning. Following [24], [26], [28], [32], [33], we use no augmentation on the images. A validation set with 6,000 images is split from the training set. We select the model with the lowest validation error during training and report the test error. The pixel values are divided by 255 so they are in the [0, 1] range.

B. System Specifications

All the experiments were run on a system with Nvidia Titan XP GPU with an Intel(R) Xeon(R) Platinum 8260 CPU and 264 GB RAM. Training and testing of networks was done on the GPU.

C. Training

We train all our networks following [1]. All the networks were trained using stochastic gradient descent (SGD). We use a batch size of 64 and train networks for 300 and 40 epochs for CIFAR10 and SVHN respectively. The initial learning rate is set to 0.1, which is divided by 10 at 50% and 75% of the total number of epochs. We use a weight decay of 10^{-4} and a Nesterov momentum of 0.9 without dampening.We use the weight initialization introduced by [34].

D. Numerical experiments

We train and test a DenseNet model of the same configuration for four different types, that are 100 layer BC with the initial convolution of kernel size 7x7, 100 layer BC with the initial convolution of kernel size 3x3, 40 layer BC with the initial convolution of kernel size 7x7 and 40 layer BC with the initial convolution of kerne size 3x3. Here BC denotes the bottleneck-compression variant of DenseNet. All the models are trained with a growth rate of 12. The custom function is used to specify the connectivity in the dense blocks. This function is used only in the last dense block then on the second and third dense block and lastly on all dense blocks. Lastly, we add intermediate weights between connections. These weights are added in three variants, firstly a simple weight is added between each skip connection, on the second variant these weights are also added between the feed-forward connections.

	Model	Top1 error	Top5 error	Paraeters	Flops(MACs)
100 BC	Default	5.49	0.18	769162	296523584
	last1custom	5.51	0.19	608362	286232384
	last2custom	6.63	0.2	476962	252593984
	all custom	6.89	0.21	404362	178251584
40 BC	Default	7.14	0.21	176122	74404176
	last1custom	8.22	0.31	160522	73405776
	last2custom	8.48	0.35	147322	70026576
	all custom	7.96	0.23	138922	61424976

TABLE I
EXPERIMENTAL RESULTS ON CIFAR10 DATASET WITH INITIAL CONV $3 * 3$

	Model	Top1 error	Top5 error	Parameters	Flops(MACs)
100 BC	Default	10.89	0.47	772090	19413120
	last1custom	10.9	0.5	611290	18769920
	last2custom	11.55	0.59	479890	16667520
	all custom	13.5	0.58	407290	12021120
40 BC	DenseNet	13.28	0.73	179050	5526720
	last1custom	13.54	0.57	163450	5464320
	last2custom	13.96	0.64	150250	5253120
	all custom	14.53	0.64	141850	4715520

TABLE II
EXPERIMENTAL RESULTS ON CIFAR10 DATASET WITH INITIAL CONV $7 * 7$

Abbreviations: *Default:* Original DenseNet, *last1custom:* Root-n connectivity in the last dense block, *last2custom:* Root-n connectivity in the second last dense block, *all custom:* Root-n connectivity in all dense blocks

In the last variant, channel-wise weights are added between each skip connection.

V. Results and Discussions

A. Discussion of accuracy

DenseNet-100: Table I shows the results obtained on the Cifar-10 dataset for the 100 layer DenseNet with initial $3 * 3$ convolution, bottleneck layer (B) and compression factor (C) of 0.5. We observe that the custom connection of root-n in the last dense block increases the top-1 and top-5 error by 0.02% and 0.01% respectively as compared to the original DenseNet. This maybe due to the presence of noisy and redundant skip connections. The root-n connectivity ignores these connections and retains informative skip connections, thereby, not having much effect on the classification accuracy even on reduction

	Model	Top1 error	Top5 error	Paraeters	Flops(MACs)
100 BC	Default	3.519	0.3995	769162	296523584
	last1custom	3.8837	0.4994	608362	286232384
	last2custom	3.9720	0.5608	476962	252593984
	all custom	3.745	0.5109	404362	178251584
40 BC	Default	3.753	0.4264	176122	74404176
	last1custom	4.153	0.461	160522	73405776
	last2custom	4.191	0.5532	147322	70026576
	all custom	3.995	0.484	138922	61424976

TABLE III
EXPERIMENTAL RESULTS ON SVHN DATASET WITH INITIAL CONV $3 * 3$

Abbreviations: *Default:* Original DenseNet, *last1custom:* Root-n connectivity in the last dense block, *last2custom:* Root-n connectivity in the second last dense block, *all custom:* Root-n connectivity in all dense blocks

Model	Error
100BC default densenet	5.74
100BC default densenet inter weight	5.34
100BC default densenet inter weight all	5.22
100BC default densenet inter weight by channel	5.12
40BC default densenet	7.14
40BC default densenet inter weight	7.27
40BC default densent inter weight all	7.81
40BC default densenet inter weight by channel	5.74

TABLE IV
EXPERIMENTAL RESULTS OF DIFFERENT WEIGHTED MODELS ON CIFAR10 DATASET WITH INITIAL CONV $3 * 3$

Abbreviations: *Default:* Original DenseNet, *inter weight:* One learnable weight between each skip connection, *inter weight all:* One learnable between each skip and feed-forward connection, *inter weight by channel:* One weight for each channel between each skip connection

of about 1 million parameters. This phenomena can also be observed from the heatmap in fig. 2(c). The heatmap shows that distribution of absolute values of weights on the skip connections that were trained through the backpropagation algorithm. From this, we observe that there are many low weight skip connections that can be considered as uninformative.

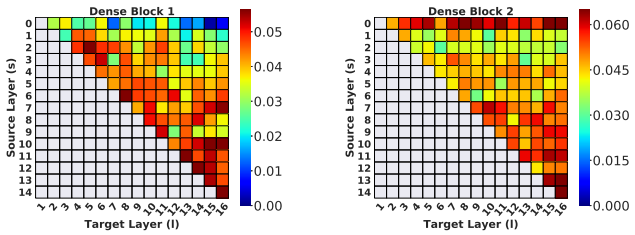
The custom root-n connectivity in the last two dense blocks increases the top-1 and top-5 error by 1.14% and 0.02 %. This maybe due to the fact that many more important skip connections are not being considered which reduces the representational capability of the model. We can also observe from the 2(b) that the magnitude of the weights in the second dense block is more uniformly distributed as it doesn't contain too many spikes or lows. This observation in the heatmap means that most of the skip connections in the second dense block provide non redundant information to the classifier becaause of which they get uniform magnitude of weights.

The custom root-n connectivity in all the three dense blocks further increases the top-1 and top-5 error by 1.14% and 0.03%. This maybe due to the fact that not considering many connections from the first dense block further reduces the representational capacity of the model. We can see from the heatmap 2(a) that the skip connections of the first dense block also have considerable magnitude as we increase the layer number within the block. This means that many of the skip connections of higher layers inside the first dense block provide useful information to the model.

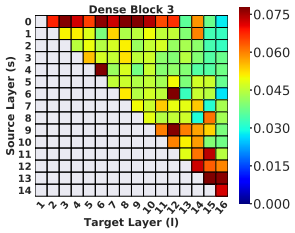
Table II shows the results obtained on the Cifar-10 dataset for the 100 layer DenseNet with initial $7 * 7$ convolution and BC. We observe that the overall accuracy is lesser then the DenseNet with initial $3 * 3$ convolution for every experiment. This is due to the large kernel size for small resolution images $(32 * 32)$ that masks the information content of small valued pixels due to the presence of large valued pixels. Moreover, we observe a similar aforementioned pattern of accuracy.

DenseNet-40: The 40 layer DenseNet shows lesser accuracy as compared to the 100 layer DenseNet. This is due to the less representational capability of the shallower DenseNet.

Weighted connections: We can observe from table IV that there is a significant affect of weights introduced between the connections. We observe that for the 100BC case, introduction



(a) block1 (b) block2



(c) block3

Fig. 2. Heatmap of weights on the skip connection of DenseNet-100 BC.

of one weight for each skip connections one separate weight for each skip connection results in an reduction in accuracy of 0.40%. We also observe that the introduction of weights in all the connections which includes skip connections as well as feed forward connections results in an reduction in error of 0.52%. We also observe that the introduction of channel wise weights in each skip connection results in the best performance and gives a reduction in 0.62% in error. This maybe due to the fact that the learnable weights provide enough linear transformation to the original values to include more diversity in the model which helps us in increase the accuracy of the model.

For the 40BC case, we can observe that the error rate of original DenseNet is 7.14%. There is a minor change in trend here as introduction of weights increase the error by 7.27% which is an increment of 0.16%. The introduction of weights in all the connections including skip and forward connections results in an increase in error of 0.61%. However, the introduction of channel-wise weights gives the best performance with a reduction in error rate of 1.4%. This maybe due to the channel-wise weights which act as dampening factors for noisy and unimportant channels.

B. Discussion of Heatmap

We observe from figure I the heatmap of the weights introduced in the skip connections. From the heatmap of 2(a) we can observe that the 0th layer skip connections have very less weights. And the weights of the skip connections improve/increase as we go towards the higher layers within the dense block. That is the higher layers like 8-16 have more weightage on the skip connections as compared to the lower layers like 1-7. We can say that the more abstract features are being becoming more important when being skipped as compared to the lower complexity features.

From the heatmap of 2(b) we can observe that the 0th layer skip connections have high magnitude of weights. This maybe due to the fact that the features coming from the dense 2(a) go through a transition layer and thus, contain a considerable amount of low and high complexity features. This diversity causes the weights to be more on the skip connections of 0th layer. This diversity is given high importance when being skipped in the second dense block. We can also see that the higher layer skip connections are again given more weightage. As also seen in the dense block1. The skip connections weights of other layers (lower layers) get an averagely distributed weightage. Again we can see that more abstract features are becoming more important when being skipped.

From the heatmap of dense 2(c) we find that the 0th layer skip connections again have a good amount of weightage. This again is similar to the phenomena observed in the dense block2 wherein the diverse features of layer 0 are being highly helpful in classification for the consecutive layers when being skipped. We also see that the higher layer features are highly important when being skipped. The higher layers 13-16. as compared to the lower layer skip connections. The skip connections of layers 1-12 in the dense 2(c) are having low weightage and thus depict the redundant connections. This can also be correlated with the earlier results wherein removing the connections using root-n connectivity in the last dense block doesn't drastically reduce the accuracy.

VI. Conclusions and future works

In this work, we present results and experiments supporting our hypothesis that not all of the connections in a DenseNet model provide important information for classification. The proposed connectivity pattern of Root-n shows that there can be a significant saving of FLOPS with minimal reduction in accuracy. The proposed approach of introducing weights in different types of connections aided in comprehending the importance of features of each layer during skipping. This also provided a new insight into the working of DenseNet as we perceived the working of different dense blocks.

The future works include robust methods for pruning the unimportant connections and for introducing more diversity in model.

References

[1] 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*, 2017, pp. 4700–4708.

[2] S. Sengupta, S. Basak, P. Saikia, S. Paul, V. Tsalavoutis, F. Atiah, V. Ravi, and A. Peters, "A review of deep learning with special emphasis on architectures, applications and recent trends," *arXiv preprint arXiv:1905.13294*, 2019.

[3] 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*, 2016, pp. 770–778.

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

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

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[7] 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*, 2018, pp. 2752–2761.

[8] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in neural information processing systems*, 2015, pp. 1135–1143.

[9] S. Han, H. Mao, and W. J. Dally, "Deep compression: compressing deep neural network with pruning, trained quantization and huffman coding. corr abs/1510.00149 (2015)," *arXiv preprint arXiv:1510.00149*, 2015.

[10] Z. Mariet and S. Sra, "Diversity networks: Neural network compression using determinantal point processes," *arXiv preprint arXiv:1511.05077*, 2015.

[11] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Pensky, "Sparse convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 806–814.

[12] Y.-D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin, "Compression of deep convolutional neural networks for fast and low power mobile applications," *arXiv preprint arXiv:1511.06530*, 2015.

[13] T. Rui, J.-c. Fei, P. Cui, Y. Zhou, and H.-s. Fang, "Head detection based on convolutional neural network with multi-stage weighted feature," in *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*. IEEE, 2015, pp. 147–150.

[14] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710*, 2016.

[15] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Advances in neural information processing systems*, 2016, pp. 2074–2082.

[16] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "Eie: efficient inference engine on compressed deep neural network," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 243–254, 2016.

[17] S. Srinivas, A. Subramanya, and R. Venkatesh Babu, "Training sparse neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 138–145.

[18] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1389–1397.

[19] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2736–2744.

[20] S. Anwar, K. Hwang, and W. Sung, "Structured pruning of deep convolutional neural networks," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 13, no. 3, pp. 1–18, 2017.

[21] 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.

[22] X. Xiao, S. Lian, Z. Luo, and S. Li, "Weighted res-unet for high-quality retina vessel segmentation," in *2018 9th International Conference on Information Technology in Medicine and Education (ITME)*, Oct 2018, pp. 327–331.

[23] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

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

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

[26] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[27] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *arXiv preprint arXiv:1412.6550*, 2014.

[28] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Artificial intelligence and statistics*, 2015, pp. 562–570.

[29] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[30] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in neural information processing systems*, 2015, pp. 2377–2385.

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

[32] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," *arXiv preprint arXiv:1302.4389*, 2013.

[33] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional neural networks applied to house numbers digit classification," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, 2012, pp. 3288–3291.

[34] 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*, 2015, pp. 1026–1034.