ISSN (e): 2250 – 3005 || Volume, 06 || Issue, 4|| April – 2016 ||
International Journal of Computational Engineering Research (IJCER)

An Efficient Convolutional Neural NetworkArchitecture for Mobile Devices

MonalisaSamal [1],Sudhir Kumar Sa[2] ,D.K.Mohanty[3]
[1] *Assistant Professor, Department of Electronics and Communication Engineering, Gandhi Institute For Technology (GIFT), Bhubaneswar*
[2]*Assistant Professor, Department of Electronics and Communication Engineering, Gandhi Engineering College,*
[3] *Assistant Professor, School of Computer Engineering , KIIT University ,Bhubaneswar*

Abstract
Deep NeuralNetworks (DNNs) have achieved great success in many machine learning tasks over the last decades.DNNs perform better than classical machine learning methods on various classification problems by producing good quality feature maps through successive convolution operation(s). However, when implementing a DNN in an embedded system or SoC for mobile devices, there can be a significant burden on the internal memorydesign due to the presence of large size parameters. In this paper, we propose a new DNN that reduces computation time and the number of model parameters but performsreasonablygood. This paper proposes an efficient CNN structure that shows satisfactory performance even with only a small numberofconvolutionfilters. Here, we propose a MacroUnit (MU) to reduce heavy computations and to learnvariousfeaturemaps, then an asymmetric convolution of the well-known Inception networkisemployed to further efficiently manipulate feature maps within the MU. Finally, allthefeaturemapsproducedfromMU(s)ofeachlayerare concatenated and then the grouped feature map is distributed to all the MUs of the next layer for transferring richer information. Experimental results show that the proposed network achieves about 10% higherperformance than DenseNet-BC in case of extremely small parameter size for CIFAR-100. The proposed network also has very few learning parameters and smaller floating point operations per second (FLOPS) than the other networks optimized for mobile devices such asMobileNet.

I. INTRODUCTION

Deep Neural Network (DNN) can outperform many conventional machine learning methods, due to the increased amount of training data, more powerful computing resources as well as dramatically increased model parameters. Recently, DNN techniques [1-3] that can be considered as a catalyst for this deep learning era have been devised. The end-to-end learning optimizes all processing steps simultaneously, and it leads to better performance and smaller systems. Because of outstandingperformance beyond traditional machine learning techniques,deeplearningisexpandingitsscopetoavariety ofapplications,includingcomputervision,medical,andarchitecturalapplications.AlexNet[4]isknownasthemost famous convolutional NN (CNN) among early deep learning methods.

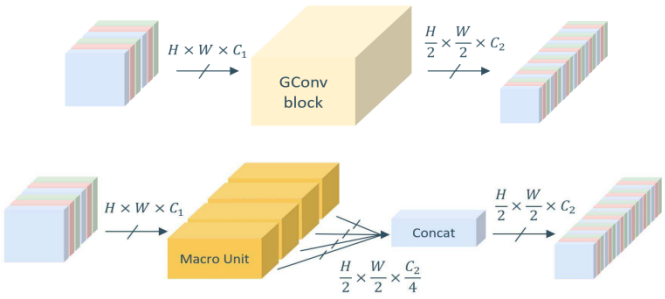


Figure 1: The main concept of the proposed method. *Up*: Learning scheme of group convolution block. *Down*: Learning scheme of proposed convolution block. Here *H* ,*W*represent

the horizontal and vertical sizes of the feature map, respectively and C_n represents the number of channels.

Next, competition to develop deeperand better CNNs than AlexNet has become fierce, and so the well-known visual geometry group (VGG) network was born out of such competition [5]. However, since VGG, many researchers have encountered limitations in increasing network depth. He et al. caused sensation by presenting ResNet [7], which enables a deeper structure of the network using residualconnections.

ResNeXt [8] and Xception [9] have also been proposed,whicheffectivelylearnsthefeaturemapbyfocusingonthe convolution filter configuration rather than the feature map connection. Recently DenseNet [10], which is a network with differentiating connection methods between feature maps, based on the existing ResNetbecamefamous.

On the other hand, extensive size of learning parameters and huge computational complexity make it difficult to

implement existing CNNs with embedded software or SoC for mobile devices, drones, and social robots. To solve this problem, CNNs focusing on sparcity connection aimed at hardware-friendly structure have been developed [13, 14].

However, conventional CNNs, including [13] and [14], still have significant computational complexity and parameter size, making them cumbersome to implement. The fundamental problem of CNNs in terms of computational complexity and memory size is the use of so-called group convolution (GConv) [4], There are some cases, whichcausesunnecessaryincreaseofparameter size and computational complexity [26,35].

To solve the above-mentioned problem, MobileNet [16] and ShuffleNet [17] have adopted factorized convolution such as depth-wise separable convolution [6].Nonetheless, the tradeoff between performance and parameter size/computation is still an unresolved issue in theresearch on CNN. The reason is that even if you use factorized convolution,manyfiltersperconvolutionoperationarestill required. This can eventually lead to a redundancy in learning high dimension feature map[31].

This paper proposes an efficient CNN structure that shows satisfactory performance even with only a small numberofconvolutionfilters.AsshowninFig.1,learningafeaturemapbydividingasingleGConvblockintomultiplemac rounits(MUs)isakeyconceptoftheproposednetwork. Each MU consists of much smaller number of convolution filters than GConv block, and furthermore it adopts separable convolution of Inception-V4 [15], which has considerable advantages in terms of parameter size and computation amount. Meanwhile, we propose a method to simultaneously concatenate feature maps generated from multiple MUs of each layer and distribute them to the next layer, as shown in Fig. 2, so that a rich feature map can be generated even with MUs composed of a small number of convolution filters. Therefore, the proposed network can learn sufficiently rich feature maps with relatively a small number of convolutionfilters.

We verified the classification performance of the proposed MU-based network (MUNet) on highly competitive benchmarking datasets, i.e., CIFAR, SVHN, and Tiny-ImageNet. For example, for CIFAR-100 dataset, MUNetshowsaperformanceimprovementofupto10%in a small parameter region compared to the DenseNet-BC. Also, in the reasonably similar error rate range, MUNetexhibits more than 15 times less parameter size and fewer FLOPS thanMobileNet.

II. RELATED WORKS

Novel convolution filter. Inception-V4 is an inception module that combines the results from multiple filters of differentsizes.Toreducethecomputationalcomplexity,the inception module adopts the asymmetric convolution instead of the two-dimensional (2D) convolution. Also, ResNeXt and Xception networks have shown that feature mapscanbeefficientlylearnedbyimplementingpoint-wise convolution and depth-wise convolution based on (1,1) convolution filter and (3,3) convolutionfilter.

Connection between feature maps. As mentioned earlier, ResNet is a way to increase the number of parameters and pursue a deeper network than AlexNet and VGG. Adding the by-pass connection to the existing VGG networkisthekeyidea,whichsolvedthegradientvanishingproblemandhelpedthesmoothtransmissionoffeaturemap information. Thereafter, modified versions of ResNetappearedinwhichvariousmethodswereappliedtoResNet [18,19].

Beyond the ResNet family of algorithms, DenseNet has introduced two new concepts. First, DenseNet utilized residual connections of concatenate style rather than those of summation style, thus enabling rich learning of feature maps.Second,DenseNethasmadeconnectionsbetweenall the layers of the block module so that feature map and gradientinformationcanbeseamlesslyconveyedacrosstheentirenetwork.Inaddition,FractalNet[11]isremarkablein that it builds up a deep network without residual connections and boosts overallperformance.

Efficient Model Designs. MobileNet is based on depth-wise separable convolution which factorize a standard convolution into a depth-wise convolution and a 1x1 convolution called a point-wise convolution. Note that conventionalconvolutionperformsfilteringandcombining simultaneously. The depth-wise separable convolution

of MobileNet splits this into two layers, i.e., one layer for filtering and the other layer for combining. So such a factorization has the effect of drastically reducing computationandmodelsize.Therefore,MobileNetenables very fast inference time while showing reasonable performance.
ShuffleNet has a more advanced convolution structure than MobileNet. ShuffleNet is also based on a depth-wise separable convolution, but realizes a small network with few channels through the channel shuffle technique. So ShuffleNet has similar performance to MobileNet, but its parameter size and computation becomes smaller than that of MobileNet.
On the other hand, NASNet model [20], which has been attracting attention recently, is not a network designed to haveafundamentallysmallparametersize,sowedonotuseNASNet as a benchmarking network of MUNet in this paper.

III. APPROACH

As shown in Fig. 2, MU is a crucial component of the proposed network. Feature maps learned in independent MUsofeachlayerareconcatenatedanddistributedtoMUs of the next layer. After repeating the above steps in several layers as shown in the figure, the final feature map is produced and then global pooling is applied. Finally, the vector information is obtained through the fully connected (FC) layer of the same level as the class ofthe

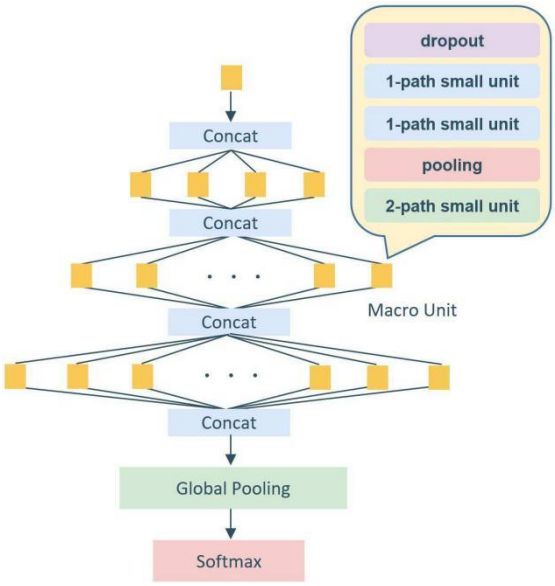


Figure 2: Overall framework of the proposed network.

corresponding data set, and the softmax function is taken and the final prediction is performed.

3.1 Macro unit

ThissectiondescribesthestructureoftheMU,i.e.,akey element of the proposed network. The overall MU configuration is shown in Fig. 3(a). After the network regularization process such as dropout [12], feature map learning is performed through multiple 1-path small units. The primary function of the 1-path small unit(s) is to increasethenumberofinternallayersintheMUtoprevent performance degradation. We then reduce the size of the featuremapthroughthepoolingprocess.Finally,the2-path small unit is applied. Since the 2-path small unit is constructed based on the dynamic connections of 1D convolutions in different directions, which were also proposed in the inception module of [15], and symmetric convolution, it outputs feature maps with more abundant characteristics than 1-path small unit do. As a result, the MUNet having a small number of convolution filters can significantly reduce the computational complexity and the parameter size compared to the conventional GConv-based networks.

3.2. Small units

As in Fig. 3(a), the MU uses two kinds of small units. First, the 1-path small unit is used for learning a richer feature map by increasing the number of layers in the MU, and has the detailed structure of Fig. 3(b). The number of 1-path small units in the MU is considered a hyper parameterofthenetwork,suchasdropout,whichiscoveredin Section 4.6. 1D convolutions of (1, N) and (N, 1) are performed after the 2D convolution of (N, N), and then the results are summed as shown in the figure. As

mentioned above, the reason for using asymmetric convolution filters suchas(1,N)and(N,1)istoreducethecomputationalcostoftheconvolutionoperation.Thecomputationalcostofthe standard 2D convolution is asfollows.

$$C_1 \times C_2 \times N \times N \times D_1 \times D_2 \quad (1)$$

whereC_1 and C_2 represent the number of input channels and the number of output channels, respectively, and $N \times N$ represents the convolution kernel size, and $D_1 \times D_2$ indicates the feature map size. The ratio of the computational complexity of the asymmetric 1D convolutions to the 2D convolution computation of Eq. (1) is calculated as follows.

$$\frac{C_1 \times C_2 \times (N+N) \times D_1 \times D_2 \_ 2N}{C_1 \times C_2 \times N \times N \times D_1 \times D_2 \quad N^2} \quad (2)$$

That is, the parameter reduction effect isabout $\frac{2}{N}$times.

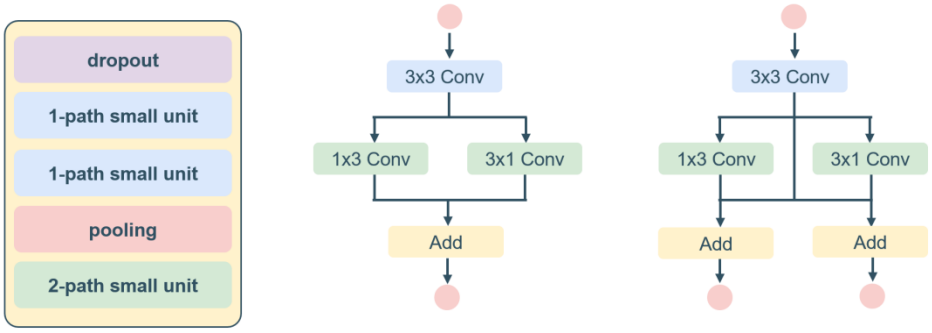
Thestructureofthe2-pathsmallunitisshowninFig.3(c). The different thing from 1-path small unit is the number of output paths. The 2-path small unit has two output paths. Eachoutputistoplustheasymmetric1Dconvolutionresult and the standard 2D convolution result. Feature maps learned by 1D convolution with different directions, i.e., asymmetric convolutions, are transferred to the MUs of the next layer, so that it is possible to generate feature maps with more abundantcharacteristics.

3.3. Comparison of GConv Block and MU Block

In this section, we compare the computational complexities of the GConv block and the corresponding MU-based block in Fig. 1. The specifications of the two network architectures are shown in Table 1. The computation cost of the GConv block is obtained from Table 1 and Eq. (1), which is as follows.

$$\begin{aligned} &Conv1 \quad : C_1 \times C_2 \times 3^2 \times D_1 \times D_2 + \\ &Conv2,3,4 : (C_2 \times C_2 \times 3^2 \times D_1 \times D_2) \times 3 \quad (3) \\ &Conv5,6 \quad : (C_2 \times C_2 \times 3^2 \times D_1 \quad |2 \times D_2 \ |2) \times 2 \\ &\qquad C_2 \times C_2 \times 3^2 \times D_1 \times D_2 = \alpha \quad (4) \end{aligned}$$

If the common term α of Eq. (4) is substituted, the total computationalcomplexityoftheGConvblockisexpressed by Eq.(5).



(a) Macro Unit (b) 1-path small unit (c) 2-path small unit

Figure 3: Architecture of macro unit (MU). (a) Overall architecture. (b) 1-path small unit. (c) 2-path small unit.

$$\left(\frac{7}{2}+\frac{C_1}{C_2}\right)\alpha \qquad (5)$$

The computational complexity of the MU-based block corresponding to the GConv block is shown in Eq. (6).

$$\left(\frac{11}{64}+\frac{C_1}{4C_2}\right)\alpha \qquad (6)$$

Based on Eqs. (5) and (6), we can find that the MU-based blockhasabout70%reductionincomputationcostoverthe GConv block, assuming that the number of convolution filters doubles in the general CNN layerhierarchy.

3.4. Interface of concatenation anddistribution

Feature maps generated from several MUs of each layer are concatenated and uniformly distributed to the MUs of thenextlayer,asshowninFig.2,sothatMUswithasmall number of convolution filters can learn rich feature maps sufficiently. For example, at the bottom of Fig. 1, we can observethatthefourMUoutputsareconcatenatedandthen delivered to the nextlayer.

Therefore,MUNethasanadvantagethatithasverysmallparametersizeandcomputationamountwithcomparabletoGConv-based networks.

3.5. Implementation details

AsshowninFig.2,theproposednetworkconsistsoffour MU-layers. The number of MUs from the top MU-layer is set to 1, 4, 8, and 16, respectively. The first MU-layer has onlyoneMU,whichiscomposedofonly2-pathsmallunit, not dropout and pooling layer considering the input characteristics of the network. The second and third MU-layers have the structure shown in Fig, 3(a). Since the last fourth MU-layer no longer needs to increase the depth ofthenetwork,weonlyusethe1-pathsmallunitonceafter the dropout.

Layer	GConv block	MU block
Input	$H\times W\times C_1$	$H\times W\times C_1$
Conv1	(3, 3) conv.	(3, 3) conv.
Conv2	(3, 3) conv.	(1, 3) & (3, 1) conv.
Conv3	(3, 3) conv.	(3, 3) conv.
Conv4	(3, 3) conv.	(1, 3) & (3, 1) conv.
Pooling	(2, 2) max pool	(2, 2) max pool
Conv5	(3, 3) conv.	(3, 3) conv.
Conv6	(3, 3) conv.	(1, 3) & (3, 1) conv.
Concat	-	Filters of 4 MUs
Output	$\frac{H}{2}\times\frac{W}{2}\times C_2$	$\frac{H}{2}\times\frac{W}{2}\times C_2$

Table 1: Network architecture between GConv block and MU.

Here(3,1)&(3,1)conv.indicatesasymmetricconvolutions.BothGConv block and MU assume C_1input feature maps.C_2 convolution filters are used for GConv block and $C_2$4 convolution filters are used for each MU. After pooling, the feature map size is halved.

Meanwhile,maxpoolingisusedinthepoolingstep,andthedropoutratesoftheremainingMU-layersexceptforthe first MU-layer are 0.2, 0.5, and 0.5,respectively.

IV. EXPERIMENTALRESULTS

We experimentally demonstrated the classification performance of MUNet for CIFAR, SVHN, and Tiny-ImageNet datasets, which are widely used as benchmarks for classification problems. We also verified the model adaptability beyond classification task.

Section4.1discussesthedatasetsusedintheexperiments anddetailsofthelearningmethod.Section4.2comparesthe performance of the proposed MUNet against the latest techniquesviaclassificationproblem.Section4.3compares the computational complexity as well as the memory cost. Section 4.4 verified the adaptability of theproposed

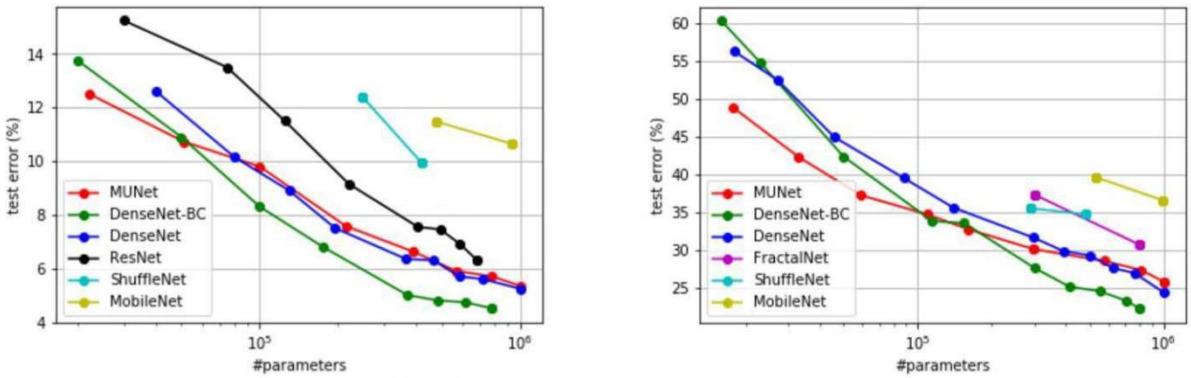


Figure 4: Comparison in terms of the parameter efficiency for CIFAR datasets. *Left*: C10+. *Right*: C100+.

network to the surrounding environment. Section 4.5 experimentally justified the structural characteristics of the MUNet. Finally, Section 4.6 shows how to choose the optimal hyper-parameters of MUNet.

4.1 Datasets and training details

Forfairevaluationoftheproposedscheme,weusedfour popular datasets: CIFAR [23], SVHN [24], Tiny ImageNet [25], and down-sampled ImageNet [33]. The latter two datasets are from ILSVRC-2012 [34]. In order to evaluate the classification performance, CIFAR, SVHN, and Tiny ImageNetareused.ThereasonforincludingTinyImageNethereistoprovethatMUNetisanetworkthatissufficiently applicable to the actual environment.Down-sampled ImageNetwasadoptedtoverifyfinetuningperformanceof the MUNet in Section4.4.

As the augmentation technique of the datasets, the onlyhorizontalmirroring and translation (uniform offsets in from -4 to 4) used in [7, 10, 11] was applied.

To optimize the proposed network model, the stochastic gradient descent (SGD) [27] method was used. Also, the weight decay of 10^{-4} and Nesterov momentum [28] of 0.9 without dampening were employed. For weightinitialization, we adopted the distribution used in [29].

When learning with CIFAR and SVHN datasets, the total number of epochs was set to 300 and the batchsize was set to 128. When learning with the Tiny ImageNet dataset, the total number of epochs was set to 300, and the batch size wassetto256,andthedropoutwasexcludedhere.Finally, we achieved the normalization effect of the feature map during learning process by using batch normalization [30] after the activationfunction.

The performance of a network was evaluated by the average of the results of three times experiments using the weights of the network after the final learning. We also employed the Tensorflow [21] based deep learning library, and used GPU named NVIDIA GeForce GTX 1080 Ti to finish the training process.

Model	Dataset	Params	Error rate (%)
ResNet [7]	CIFAR-10	0.8M	6.97
DenseNet [10]		1.0M	5.24
DenseNet (ours)		1.0M	5.12
DenseNet-BC [10]		0.8M	4.51
DenseNet-BC (ours)		0.8M	4.41
MUNet		1.0M	5.35
FractalNet [11]	CIFAR-100	0.8M	30.71
DenseNet [10]		1.0M	24.42
DenseNet (ours)		1.0M	24.04
DenseNet-BC [10]		0.8M	22.27
DenseNet-BC (ours)		0.8M	22.01
MUNet		1.0M	25.85

Table 2: Performance comparison of various CNN methods for the parameter size of about 1M in terms of error rates.

4.2 Classification performanceevaluation

ThissectioncomparedMUNetwithstate-of-the-artCNN techniques such as DenseNet, DenseNet-BC, ResNet, FractalNet, MobileNet, and ShuffleNet in terms of classification accuracy. The experimental results of MobileNet and ShuffleNet were from our own implementations, and those of the other networks were directlyextractedfromthecorrespondingpapers.IncaseofDenseNet,becausenonumericalresultsforparametersizes of about 500,000 or less are no available in [10], we implemented the DenseNet and plotted the results in the right of Fig. 4. In order to ensure the reliability of our implementation(s), Table 2 compared it with the results from [10] for the parameter sizes of about 1M. The results from[7]and[11]werealsocomparedtogetherinthetable. Note that according to [7, 10, 11], 1 M is almost the lower bound of the parameter size where thestate-of-the-art

Model	Dataset	Params	Error rate (%)	Inference time (*Std*)	FLOPS
1.0 MobileNet	CIFAR-10	1.88M	10.03	**37.42** (0.36)	51.44M
ShuffleNet× 1 ($g = 1$)		0.42M	9.93	39.17 (0.45)	43.39M
MUNet		**0.10**M	**9.80**	41.85 (0.35)	**40.30**M
1.0 MobileNet	CIFAR-100	1.97M	36.05	**37.76** (0.19)	51.62M
ShuffleNet× 1 ($g = 1$)		0.47M	34.76	39.64 (0.18)	44.49M
MUNet		**0.11**M	**34.72**	41.39 (1.38)	**40.32**M
1.0 MobileNet	SVHN	1.88M	4.98	**96.53** (0.26)	51.44M
ShuffleNet× 1 ($g = 1$)		0.42M	4.68	102.81 (0.24)	43.39M
MUNet		**0.10**M	**4.41**	104.72 (0.30)	**40.30**M
1.0 MobileNet	Tiny ImageNet	3.17M	56.48	**45.40**(0.13)	79.33M
ShuffleNet× 1 ($g = 1$)		0.70M	56.18	59.49(0.84)	59.49M
MUNet		**0.13**M	**56.13**	59.93(0.49)	**50.83**M

Table 3: Comparison in terms of parameter size, inference time, and FLOPS assuming the same error rates.*Std*in the fifth column represents the standard deviation.

methodsprovideeffectiveperformance.Wecansaythatour implementation is reliable. In addition, we can find that MUNetshowscomparableresultstothecutting-edgeCNNs for CIFARdatasets.

Next, since MUNet pursues a network with a small parametersize,weevaluateditsperformance,focusingona region where the learning parameter size is small to some extent. Because we can say that the parameter size range below 1M is a reasonable range based on the datasets we adopted (see Fig. 4), we concentrate on the comparison for the parameter size range of less than 100,000, i.e., an intermediatepointthatwepayattentionto.AsshowninFig. 4, MUNet shows consistently superior performance to ResNet, FractalNet, and DenseNet. The performance of MUNet is somewhat lower than that of DenseNet-BC, but the smaller the parameter size, the more comparable the performance.

On the other hand, one of the points we should pay attention to is the pattern of performance change according to the parameter size. MUNet shows a much lower performancedegradationratethantheothernetworksinthe parametersizeareabelow1.0M.Especially,incaseofvery small parameter size area of less than 0.1M, the proposed network shows rather superior performance than DenseNet-BC. That is, when comparing MUNet and DenseNet-BC in a very small parameter size area, MUNet has a performance advantage of up to 1.5% based on CIFAR-10 and up to 10% based on CIFAR-100. For CIFAR-100, FractalNet showed an error rate of about 37% atparametersizeof0.3M,whileMUNetshowedabout37% error rate at parameter size of 0.059M. In other words, assuming the same classification accuracy, it is sufficient that MUNet is as small as 1/5 of the parameter size than FractalNet.

In addition, we can observe from Fig. 4 that MobileNetandShuffleNet perform significantly poorer than the other

networksintermsofperformancevs.parametersize.ThisisbecauseMobileNetandShuffleNetarethenetworksthatare developed to reduce the amount of computation ratherthan to improve performance. In this aspect, the proposed MUNet should be noted that even if the parameter size decreases, the performance degradation rate is veryslow.

4.3 Complexity analysis

ThemainpurposeofMUNetistoderiveagoodtradeoff of performance and computation cost/parameter size to the extent that it can be implemented with embedded software or SoC for applications such as mobile devices. In this section, we analyze the computational cost and complexity of the proposed MUNet. In this experiment, the number of convolution filters in the first MU-layer of MUNet was set to 6 and the number of convolution filters of the remaining MU-layers was set to 4 for fair comparison. On the other hand,thebenchmarkingnetworksofMUNetwerelimitedtoMobileNet and ShuffleNetforconvenience.

Table 3 shows the comparison of MUNet with three networks. We compared the parameter size (3rd column), inference time (5th column), and FLOPS (6th column)when they show experimentally similar classification accuracy, i.e., error rate (4th column). Inference times were measured in units of seconds on the desktop environment embedded with the Intel® i7-7700 CPU@3.6 GHz w/ 8 processors. FLOPSindicatestheamountofcomputation,anditincludes all operations such as convolution, pooling, and even multiplication, addition, and conditional statement for activation function. The inference time is measured as the sum of all test images in thedataset.
From Table 3, we can find that MUNet has a smaller parameter size than the other networks. For example, on a

Model	Dataset	Params	Error rate (fine tuned / original) (%)	FLOPS
1.0 MobileNet	CIFAR-10	1.88M	9.70 / 10.03	51.44M
MUNet		0.11M	**9.03** /**9.54**	49.58M
		0.10M	9.15 /9.80	**40.30M**
1.0MobileNet	CIFAR-100	1.98M	**31.58** /36.05	51.62M
MUNet		0.12M	32.10 /**34.58**	49.59M
		0.11M	33.54 /34.72	**40.32M**

Table 4: Fine tuning performance comparison between MobileNet and MUNet.

Method	Error rate (%)	Params	FLOPS [Mult. Add If]
Single MU per layer	9.01	0.49M	203.72M [101.75 101.16 0.81]
Multiple MUs per layer (MUNet)	9.80	0.10M	40.30M [20.11 19.88 0.31]

Table 5: Performance comparison according to MU-layer structure for CIFAR-10.

CIFAR dataset, MUNet achieves about four times the parameter reduction in comparison to ShuffleNet.
On the other hand, while MUNet has always smaller FLOPS than ShuffleNet and MobileNet for all datasets, its inference time is slightly larger than the other networks. This is because MUNet has a small number of filters used per convolution, but it performs many convolution operations due to the nature of the network. This issue will be touched again in Section 5.

4.4 Adaptability to the surroundingenvironment

MUNet is a CNN model designed for mobile device purpose. It is important that the CNN model fitted to a specificenvironmentshouldmaintainitsperformanceeven if the surrounding environment changes. Therefore, this section verifies whether MUNet conforms to the characteristics of the mobile device through the fine tuning experiment of MUNet with very small parameters. Unlike the experiment in the previous section, the reason for excluding ShuffleNet from the benchmarking group is that thefinetuningexperimentwasnotperformedinShuffleNet. First, we trained the MUNet by using a down-sampled ImageNetdataset,andperformedafinetuningprocesswith the CIFAR dataset. Here, the reason for pre-training using down-sampled ImageNet is that as shown in [33], down-sampledImageNetproducessimilaroptimalhyper-parameterswhileexhibitingfasterlearningconvergence speed than original ImageNet. In addition, sincetheimagesizeofdown-sampledImageNetisthesame as the image size to be fine-tuned, more accurate verification of fine tuning performance is possible. In both networks,thespecificationofTable3wasusedasis.Inthe case of MUNet, the number of convolution filters of small unitswaschanged.AsshowninTable4,whenfinetuningis performed, MUNet shows an additionalperformance

The number of 1-path small units	Error rate (%)	FLOPS [Mult. Add If]
1	36.82	27.00M [13.48 13.33 0.19]
2	34.72	40.32M [20.12 19.89 0.31]
3	34.40	53.69M [26.79 26.48 0.42]

Table 6: Performance comparison according to the number of 1-path small units for CIFAR-100.

improvement of about 2.4% based on CIFAR-100. On the other hand, MobileNet has a performance improvement of about 4.5%. It can be said that MobileNet, which has a larger number of parameters, could learn richer feature information than MUNet when pre-trained with a huge amount of datasets. Note that despite much smaller parametersize,MUNetshowssuccessfulfinetuningresults. ThismeansthateventhoughthemodelsizeoftheMUNetis very small, it can learn meaningful feature information when pre-trained with a large amount ofdatasets.

4.5 Performance evaluation according to MU-layer structure

The core concept of MUNet is to use multiple MUswith a small number of convolution filters at each MU-layer. In this experiment, the structural characteristics of such a MUNet are verified (see Table5).

The first row of Table 5 shows the case where the MU-layer is designed with a single MU per layerstructure. In this case, the number of convolution filters is increased enough to compensate for the reduced number of MUs per MU-layer for fair comparison. The result was compared with authentic MUNet (see the 2nd row in Table5).

As can be seen from the experimental results in Table 5, the single MU per layer structure shows a minor accuracy improvement of 0.8% in comparison with the original MUNet, but it increases about 5 times both the parameter size and FLOPS. As a result, we can find that the architecture of MUNet is superior in performance.

4.6 Selection of the appropriatehyper-parameter

ThecoreunitsoftheMUarchitectureare1-pathsmallunit and 2-path small unit. In Section 3.3, we emphasized that the 2-path small unit plays an essential role in MUNet performance,andalsothereasonfortheexistenceof1-path smallunitistoincreasethenumberofinternallayersofMU model.Sothenumberof1-pathsmallunitscanberegarded as ahyper-parameter.

In this section, we performed an experiment to find an optimal hyper-parameter for MUNet, i.e., the best number of1-pathsmallunits.TheresultisshowninTable6.Asthe number of 1-path small units increases, the FLOPS increases linearly, but the error rate decreases sharply and saturates. Therefore, this paper adopted two 1-path small unitsperMU,pursuingoptimaltrade-offperformance.The number of MUs per MU-layer can also be regarded as a hyper parameter, although it has not been tested in this paper.

V. DISCUSSION

As can be seen in Table 3, MUNet has significantly smaller parameter sizes and smaller FLOPS than the other networks. However, in terms of inference time, MUNet is somewhat inferior to the other networks unlike in terms of FLOPS. The reason is that MUNet has a small number of filters per convolution, but it must perform many convolution operations inherently. MUNet has a more complex structure than a simple group convolution-based network when forming the whole network structure under the same conditions as [21]. The more complicated the structure, the more burden can be placed on the inference time. Therefore, we can analyze that MUNet has less FLOPS but provides somewhat longer inference time than the other networks. In order to overcome this problem, shorteningthecomputationtimeoftheconvolutionasin[22, 32] can reduce the inference time ofMUNet.

VI. CONCLUSION

In this paper, we propose a new CNN architecture that escapes from the existing group convolution structure in order to design a deep neural network suitable for mobile devices. Since the conventional group convolution-based network uses a large number of convolution filters, the amount of computation is large and there may be a considerable redundant computation. In order to solve this computational cost problem fundamentally, this paper presented a MU-based CNN using only a small number of convolutionfilterswherericherfeaturemapscanbelearned by using multiple MUs instead of a single GConv block. Experimental results show that the proposed MUNet consumes significantly less computation cost while maintaining comparable performance with existing networks including state-of-the-artCNNs.

REFERENCES

[1] R. Salakhutdinov, G. Hinton. Deep boltzmann machines. In *Artificial Intelligence and Statistics*, pages 448-455, 2009.

[2] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning*, pages 609-616,2009.

[3] Y. L. Boureau, Y. L. Cun. Sparse feature learning for deep belief networks. In *Advances in neural information processing systems*, pages 1185-1192,2008.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenetclassificationwith deep convolutional neural networks. In*Advancesinneuralinformationprocessingsystems*,pages 1097-1105,2012.

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

[6] L. Sifre. Rigid-motion scattering for image classification. PhD thesis, Ph. D. thesis,2014.

[7] 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 770-778,2011.

[8] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*,2011.

[9] F. Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv preprint arXiv:1610.02357*, 2011.

[10] G.Huang,Z.Liu,K.Q.Weinberger,andL.vanderMaaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*,201.

[11] G. Larsson, M. Maire, and G. Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXivpreprint arXiv:1605.07648*,2011.

[12] N.Srivastava,G.Hinton,A.Krizhevsky,I.Sutskever,andR. Salakhutdinov. Dropout: A simple way to prevent neural networksfromoverfitting.*TheJournalofMachineLearning Research*, pages 1929-1958,2014.

[13] X. Zhang, J. Zou, X. Ming, K. He, and J. Sun. Efficient and accurate approximations of nonlinear convolutional networks. In *Proceedings of the IEEE Conference on ComputerVisionandpatternRecognition*,pages1984-1992, 2010.

[14] 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*, pages 2074-2082, 2011.

[15] C. Szegedy, S. loffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *AAAI*. Vol. 4,2010.

[16] A. G. Howard, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*,2010.

[17] X. Zhang, et al., ShuffleNet: An extremely efficient convolutional neural networks for mobile devices. *arXiv preprint arXiv*:1707.01083,2010.

[18] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision*, pages 646-661,2011.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630-645,2011.

[20] B. Zoph, and Q. V. Le. Neural Architecture Search with Reinforcement Learning. *arXiv preprint arXiv*:1611.01578, 2011

[21] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous systems, 2010. *Software available from tensorflow. org*, 1, 2010.

[22] W. Xu, Z. Wang, X. You and C. Zhang. Efficient fast convolution architectures for convolutional neural network. In *ASIC (ASICON). IEEE 12th International Conference on*, pages 904-907,2010.

[23] A. Krizhevsky, G. Hinton. Learning multiple layers of features from tiny images. *Tech Report*,2009.

[24] I.J.Goodfellow,etal."Multi-digitnumberrecognitionfrom street view imagery using deep convolutional neural networks." *arXiv preprint arXiv:1312.6082*, 2013.

[25] A. Bastidas. Tiny ImageNet Image Classification. *url: https://tiny-imagenet.herokuapp.com/*.

[26] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv preprint arXiv*: 1510.00149,2010.

[27] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. pp. 177-186,2010.

[28] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importanceofinitializationandmomentumindeeplearning. In *International conference on machine learning*, pages 1139-1147,2013.

[29] 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 1026-1034,2010.

[30] S.Ioffe,C.Szegedy.Batchnormalization:Acceleratingdeep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448-456,2010

[31] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnor-net:Imagenetclassificationusingbinaryconvolutional neural networks. In *European Conference on Computer Vision*. pages 525-542,2011.

[32] M. Jaderberg, A. Vedaldi, and A. Zisserman. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv*:1405.3866,2014.

[33] P.Chrabaszcz,I.Loshchilov,andF.Hutter.ADownsampled Variant of ImageNet as an Alternative to the CIFAR datasets. *arXiv preprint arXiv:1707.08819*,2010.

[34] O. Russakovsky, J. Deng, H. Su, H. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211-252, 2010.

[35] S. J. Hanson, and L. Y. Pratt. Comparing biases for minimal network construction with back-propagation. In *Advances in neural information processing systems*. Pages 177-185, 1989.