Literature Survey

### 1.MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

#### <https://arxiv.org/abs/1704.04861>

Convolutional Neural Network plays an important role in computer vision. There are so many CNN based architectures AlexNet, GoogLeNet/Inception (2014), VGGNet(2014), ResNet(2015) introduced to improve the accuracy and also making the architecture more complicated but many real world applications such as robotics, self-driving car, augmented reality, the recognition tasks need to be carried out quickly .

This paper discuss about the efficient network architecture in order to build very small models that can be easily match the requirements for mobile and other embedded vision applications

MobileNet Architecture

MobileNet architecture consists of two blocks in order to reduce the no of parameters of the convolution

1.Depthwise separable convolution

2.Pointwise convolution

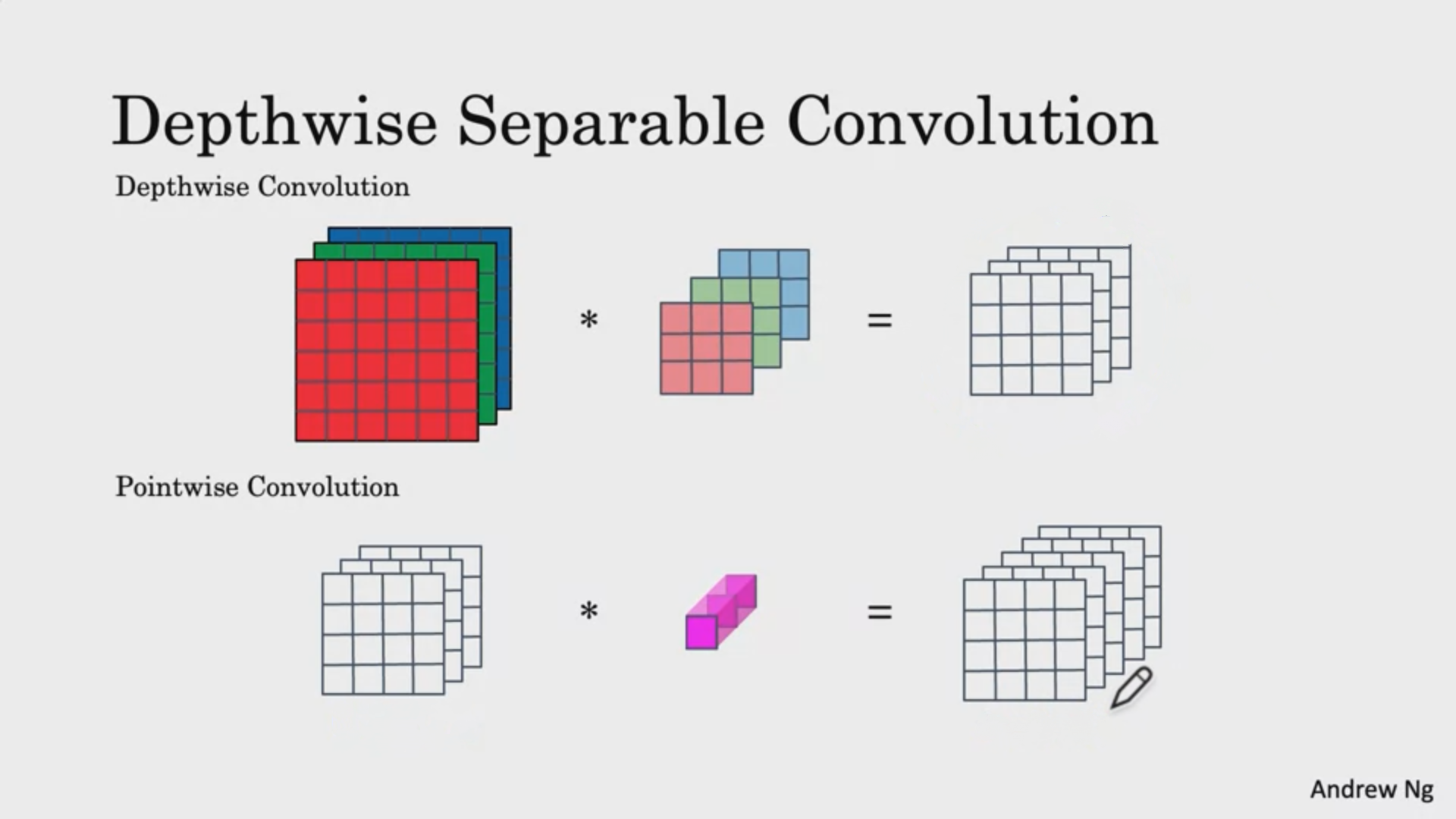
1.DepthWise convolution:

This is a type of convolution where the convolution is carried out on each channel wise instead of the the whole 3d block

This will produce an intermediate result with and the intermediate result is given for the point wise convolution

2.PointWise convolution

This is a normal convolution with a kernel size=1 which means a 1x1 convolution operator. This process can help to reduce the number of the computations in a normal convolution operator.



Mathematical representation of reducing factor :

let be the height, width,channels of the input and Be the height and width of the kernels with output channels

Total computations for the **Normal convolution** is

Computations for **Depthwise Separable Convolution** is

computations for the **Pointwise Convolution** is

**Reduction in computation**  can be given as =

### 2.High Performance Depthwise and Pointwise Convolutions on Mobile Devices

[**https://arxiv.org/abs/2001.02504**](https://arxiv.org/abs/2001.02504)

DepthWise Convolutions,pointwise Convolutions are not well utilizing the ARM processor in the mobile devices, and exhibits loss of cache misses under multi-core and poor data reuse at register level

This paper points out that the existing Depthwise Convolution and pointwise convolution implementations are poor in core scalability which is against the “more” cores in ARM processors.

Also point out that the optimization tricks suggested in the roofline article are necessary but insufficient for ARM processors.

Specifically,While both ARM and x86 processors can carry out 2 FMA(fused multiply add) instructions per cycle ARM processors can only load 1 register(from cache ) per cycle whereas x86 processors can load 4 registers per cycle

In other words, while optimizing the cache miss and increasing parallelism could eliminate the major bottleneck on x86 processors,on ARM processors those tricks could only shift the bottleneck to the traffic between the register and the cache.Based on the above observations, we therefore develop high performance version of DWConv and PWConv for mobile devices. Using techniques like loop rescheduling (Markatos and LeBlanc 1992) and register tiling (Jim´enez, Llaber´ıa, and Fernandez 2002), our implementations are able to reduce the traffic between the cache and the memory as well as the traffic between the register and the cache. Experimental results show that our implementation can respectively achieve a speedup of up to 5.5 and 2.1 against TVM (Chen et al. 2018) on DWConv and PWConv, which leads to a 46GFlops on ARM Cortex-A57 in terms of overall MobileNetV1 inference.

### 3.A Pre-defined Sparse Kernel Based Convolution for Deep CNNs

[**https://ieeexplore-ieee-org.proxylib.csueastbay.edu/document/8919683**](https://ieeexplore-ieee-org.proxylib.csueastbay.edu/document/8919683)

We use Pre Defined Sparse kernels to reduce the number of parameters in convolution neural networks.In this we exploit the structural sparsity of the input receptive-field (RF) and propose pre-defined sparse 2D kernel based CONVs to form the channels of each convolutional layer.

Suppose be the filter size of a L convolution where

be the height,width,channels of the kernel we take some values of are fixed to be zero before training and throughout the training and inference.

For example, for k = 3, there are 9 possible values of y, with y = 9 denoting the standard 2D kernel without any pre-defined sparsity, and y = 4 denotes that only four nonzero entries are allowed in the 3 3 kernel space while the remaining five entries are 0.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

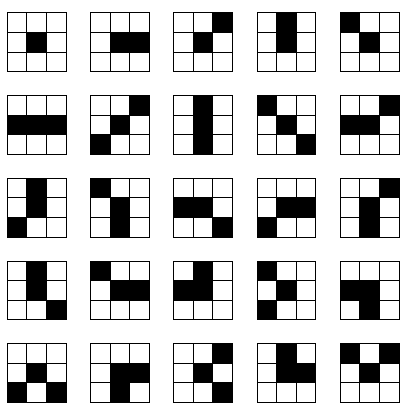
The above is an example of the kernel with blue colored values is non zero and the remainder is fixed to zero throughout the training and inference as every value of kernel is not necessary for the training and thus reducing the number of parameters of the training and total parameters of a convolutional layer.

### 4.Functionally-Predefined Kernel: a Way to Reduce CNN Computation

[**https://ieeexplore-ieee-org.proxylib.csueastbay.edu/document/8985122**](https://ieeexplore-ieee-org.proxylib.csueastbay.edu/document/8985122)

Convolution neural networks tend to consume tremendous amounts of computation and time .So there are many studies going on to reduce the computational cost.But reducing the computational cost may result in significant loss in the accuracy .

In this paper we predefine 25 kernels in the first convolution layer And the 25 kernels are determined by the **High level autocorrelation** (HLAC)



Compared to the **A Pre-defined Sparse Kernel Based Convolution for Deep CNNs** paper the main difference is instead of setting kernel to zero we predefine the kernels with a fixed value for each kernel and these are not updated in the backward pass

The black elements in the diagram represented by

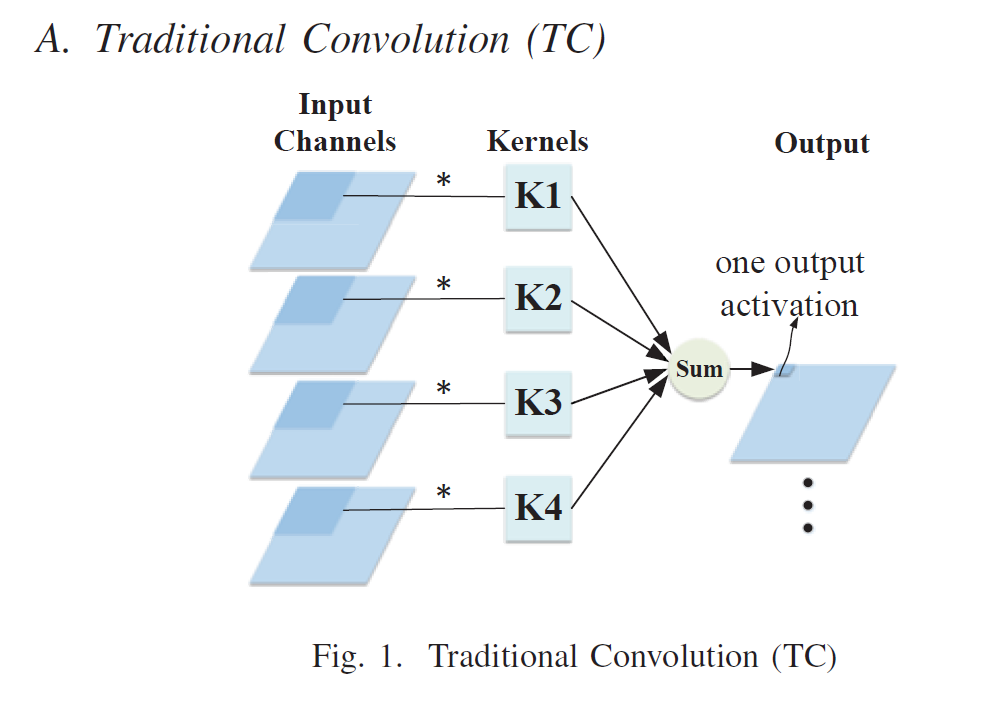
while the white elements represented by

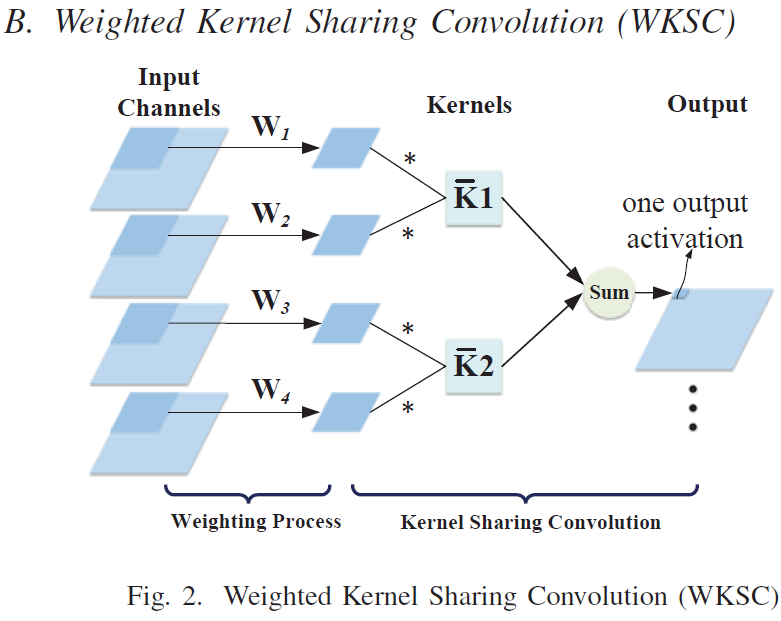
### 5.Efficient Weighted Kernel Sharing Convolutional Neural Networks

#### <https://ieeexplore-ieee-org.proxylib.csueastbay.edu/document/8698701>

This paper explains about the weighted kernel sharing , which gathers the inputs with the same kernel , so the inputs in each group can share the same convolutional kernel.

suppose if there is an image with input width,height,channels with 4 kernels k1,k2,k3,k4 and Each of size with **normal convolution** the total computation cost



suppose if there is an image with input width,height,channels with 4 kernels k1,k2,k3,k4 and Each of size with **weighted kernel sharing convolution** the total computation cost 

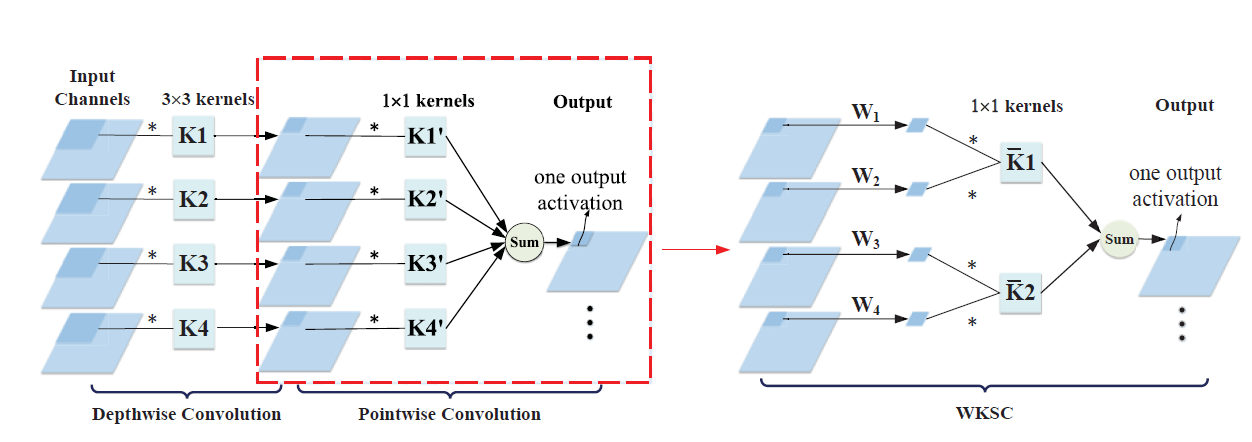
is a 2-D matrix and can be learned by back propagation and and let the size of be

this is the computation for weighting process for all of the input patch matrices

where g is the number of groups

We can observe that with relatively small-sized the weighting process takes up little computation cost and the cost of WKSC becomes

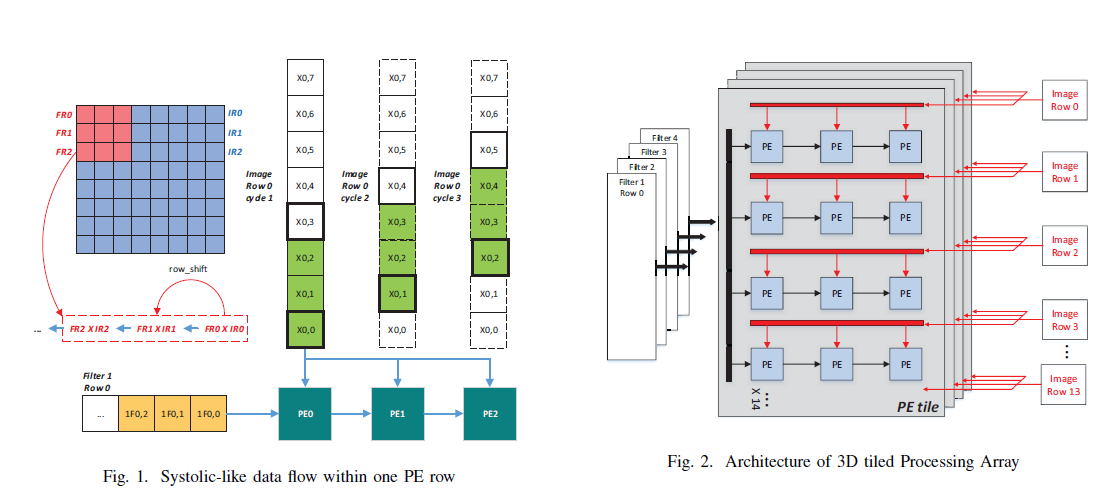
We can apply the Weight kernel sharing convolution in the mobilenet during the point wise convolutions as shown below



### 6.A 3D Tiled Low Power Accelerator for Convolutional Neural Network

#### <https://ieeexplore.ieee.org/document/8351301>

One of the challenge to run Deep learning in devices is cost of the power.This paper presents a low power accelerator for processing Convolutional Neural networks on embedded devices.This can be achieved by exploring data reuse in three different aspects in convolution operator(filters,input feature) .A systolic like data flow is proposed to the row of processing Elements which can be reused during the convolution



Reuse of input features and filters is achieved by arranging the PE array in a 3D tiled architecture,whose dimension is 3 x 14 x 4 .local storage within processing elements is therefore reduced and only cost 17.75 kB, which is 20% of the state-of-art.By adding some dedicated delay chains in each processing elements,this accelerator is reconfigurable to suit various parameter settings of convolutional layers.

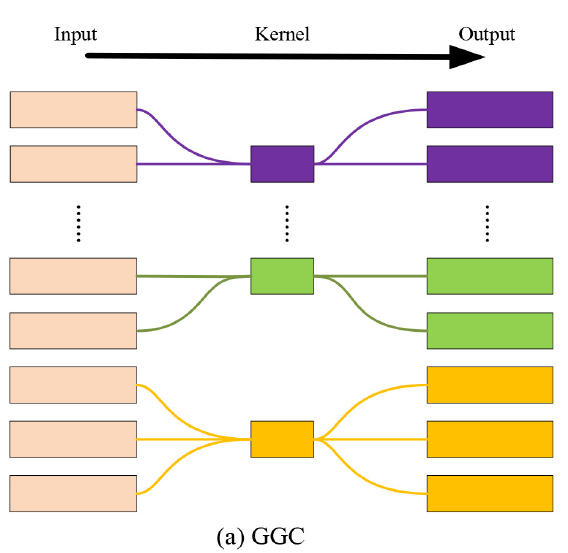
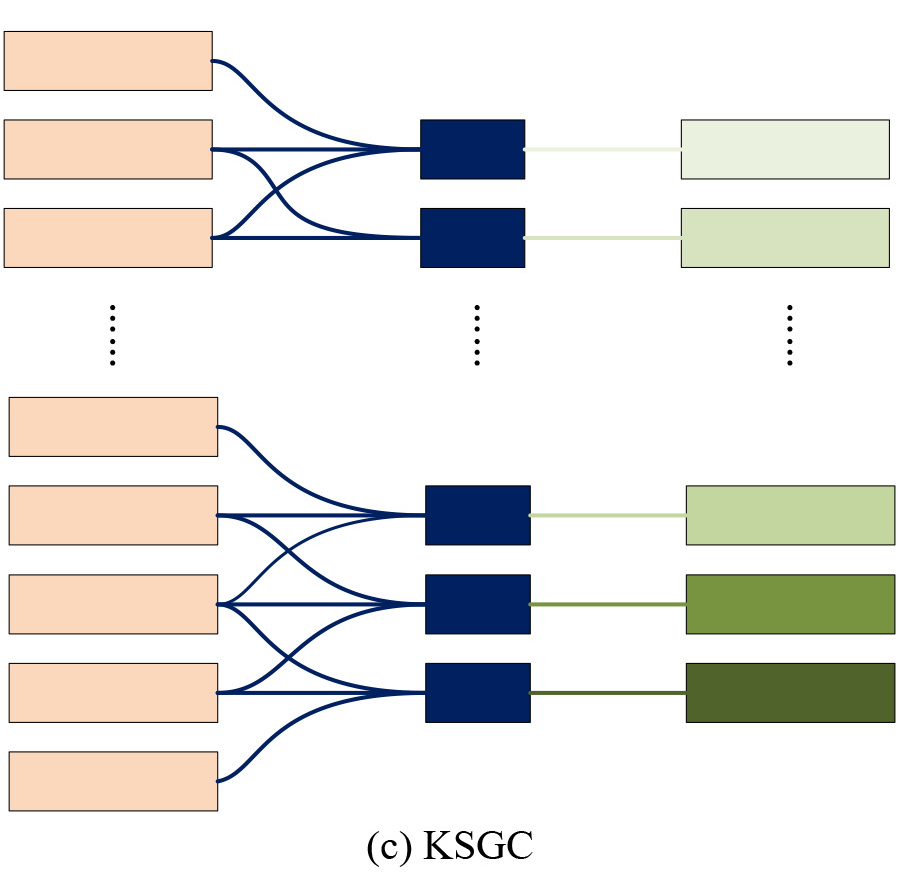
Evaluation results reveal that the proposed accelerator achieves > 92% utilization of all PEs when running the Alexnet. Compared to Eyeriss [5], our work can decrease the size of local storage by 80%, which is only 17.75 kB. Evaluated in UMC 65 nm low leakage process, the accelerator can reach a peak performance of 84 GOPS and consume only 136 mW at 250 Mhz.

### 7.Kernel Sharing in the Channel Dimension to Improve Parameters Efficiency

#### <https://ieeexplore.ieee.org/document/8941818>

This paper proposes that the convolution kernel can be shared in channel dimension by using group convolution and kernel sharing group convolution Which is named as **K**ernel **S**haring **G**roup **C**onvolution (KSCG).This is more parameter efficient than the standard convolution and general group convolution

General Group Convolution: In group convolution we see that the total input channels are separated as independent groups and each group is processed independently this reduces the number of parameters by reducing the input channels of each output channel.If each group consists of one feature map then it is depthwise convolution

Kernel sharing group wise convolution:In KSCG, different groups share the same convolution kernel.In other words the KSCG can be seen as the identical convolution kernel sliding in the channel dimension.There are overlapped input feature map between different groups

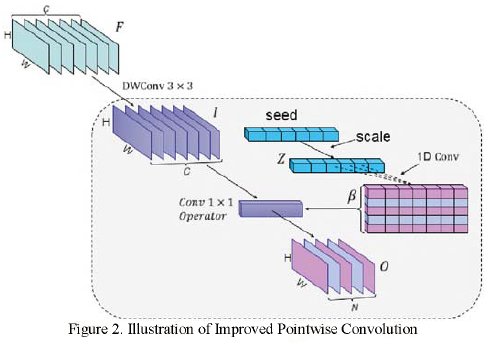
number of parameters for KSGC

### 8.An Improved Pointwise Convolutional Block for Efficient Model Compression

#### <https://ieeexplore.ieee.org/document/9040771>

Introduction of the depthwise and pointwise convolution in mobilenet is one of the improvements of the huge models like Alexnet,Resnet.With those the model is small with good accuracy.Though it plays an important role in mobilenet it is one responsible for the majority of parameters.

This paper introduces new pointwise convolution called **Improved pointwise Comvolution (IPC)** .This IPC is the block can be a substitution for regular pointwise convolution in neural network especially in compact networks like mobilenet.



In the **IPC,** after the Depthwise convolution layer as show in fig 2 , we construct a neural network to learn a mapping from seed vector where c is the number of channels of the depthwise convolution to its corresponding scoring vector .The initialization of the seed vector obeys a uniform distribution between -1 and 1.We then calculate where denotes the 1D convolution operator.In oder to adaptively evaluate channel weight according to vector Z,1D convolution are done with kernels .We will calculate

refers to activation function like sigmoid,tanh,ReLU.The main idea is to decompose to .Thus the compression ratio between the regular pointwise convolution and the improved pointwise convolution is