Supplementary Materials of Sharing Weights in Shallow Layers via Rotation Group Isomorphic Convolutions

1. Proof of $r^i(f) * K = r^i(shift^i(f * K))$

$$r^i(f) * k = r^i(f) * r^i(r^{-i}(k)) = r^i(f * r^{-i}(k)), \tag{1}$$

A kernel k can be expanded to group kernels $K = [r^0(k), r^1(k), ..., r^{n-1}(k)]$. Define $f * K = [f * r^0(k), f * r^1(k), ..., f * r^{n-1}(k)]$

$$\begin{aligned}
r^i(f) * K &= r^i(f * r^{-i}(K)) \\
&= r^i(f * [r^{0-i}(k), r^{1-i}(k), ..., r^{n-1-i}(k)]) \\
&= r^i([f * r^{0-i}(k), f * r^{1-i}(k), ..., f * r^{n-1-i}(k)]) \\
&= r^i([f * r^{n-i}(k), f * r^{n-(i-1)}(k), ..., f * r^{n-1}(k), f * r^0(k), ..., f * r^{n-1-i}(k)]) \\
&= r^i(shift^i([f * r^0(k), f * r^1(k), ..., f * r^{n-1}(k)])) \\
&= r^i(shift^i(f * K))
\end{aligned} \tag{2}$$

where $shift^1([x_1, x_2, ..., x_n]) = [x_n, x_1, ..., x_{n-1}]$,
$shift^i([x_1, x_2, ..., x_n]) = [x_{n-i+1}, x_{n-i+2}, ..., x_n, x_1, x_2, ..., x_{n-i}]$, and n is the total number of orientations.

Notably, $r^i(x*y) = r^i(x)*r^i(y)$, $r^{n+i}(x) = r^i(x)$, $shift^{n+i}(x) = shift^i(x)$. Because $r^i(\cdot)$ works on the spacial dimensions and $shift^i(\cdot)$ works on the rotation dimension, they don't influence each other and $r^i(shift^j(x)) = shift^j(r^i(x))$.

2. Proof of $F_{K_1}(r^i(f) * K) = r^i(shift^i(F_{K_1}(f * K)))$

Define $F_{K_i}(x) = [shift^0(x)*r^0(k_i), shift^{-1}(x)*r^1(k_i), ..., shift^{-(n-1)}(x)*r^{n-1}(k_i)]$, where k_i is the weight parameters of K_i.

$$\begin{aligned}
&F_{K_1}(r^i(f) * K)\\
=&F_{K_1}(r^i(shift^i(f * K)))\\
=&[shift^0(r^i(shift^i(f * K))) * r^0(k_1), ..., shift^{-(n-1)}(r^i(shift^i(f * K))) * r^{n-1}(k_1)]\\
=&[r^i(shift^{0+i}(f * K)) * r^0(k_1), ..., r^i(shift^{-(n-1)+i}(f * K)) * r^{n-1}(k_1)]\\
=&r^i([shift^{0+i}(f * K) * r^{0-i}(k_1), ..., shift^{-(n-1)+i}(f * K) * r^{n-1-i}(k_1)])\\
=&r^i(shift^i([shift^0(f * K) * r^0(k_1), ..., shift^{-(n-1)}(f * K) * r^{n-1}(k_1)]))\\
=&r^i(shift^i(F_{K_1}(f * K)))
\end{aligned} \tag{3}$$

3. Proof of $F_{K_m}(F_{K_{m-1}}(...F_{K_1}(r^i(f)*K))) = r^i(shift^i(F_{K_m}(F_{K_{m-1}}(...F_{K_1}(f*K)))))$

If $F_{K_{m-1}}(...F_{K_1}(r^i(f) * K))) = r^i(shift^i(F_{K_{m-1}}(...F_{K_1}(f * K)))$, it can be proofed easily that $F_{K_m}(...F_{K_1}(r^i(f) * K))) = r^i(shift^i(F_{K_m}(...F_{K_1}(f * K)))$. Denote $F_{K_{m-1}}(...F_{K_1}(f * K)) = M$.

Proof:

$$
\begin{aligned}
&F_{K_m}(F_{K_{m-1}}(...F_{K_1}(r^i(f) * K))) \\
=&F_{K_m}(r^i(shift^i(M))) \\
=&[shift^0(r^i(shift^i(M))) * r^0(k_m), ..., shift^{-(n-1)}(r^i(shift^i(M))) * r^{n-1}(k_m)] \\
=&[r^i(shift^{0+i}(M)) * r^0(k_m), ..., r^i(shift^{-(n-1)+i}(M)) * r^{n-1}(k_m)] \\
=&r^i([shift^{0+i}(M) * r^{0-i}(k_m), ..., shift^{-(n-1)+i}(M) * r^{n-1-i}(k_m)]) \\
=&r^i(shift^i([shift^0(M) * r^0(k_m), ..., shift^{-(n-1)}(M) * r^{n-1}(k_m)])) \\
=&r^i(shift^i(F_{K_m}(M))) \\
=&r^i(shift^i(F_{K_m}(F_{K_{m-1}}(...F_{K_1}(f * K)))))
\end{aligned}
\tag{4}
$$

According to the above proof and $F_{K_1}(r^i(f) * K) = r^i(shift^i(F_{K_1}(f * K)))$ in section 2, it can be deduced that $F_{K_m}(F_{K_{m-1}}(...F_{K_1}(r^i(f) * K))) = r^i(shift^i(F_{K_m}(F_{K_{m-1}}(...F_{K_1}(f * K)))))$.

4. More experiments

4.0.1. VGG16

The VGG16 has 16 convolution layers with kernels of 64, 64, 128, 128, 256, 256, 256, 256, 512, 512, 512, 512, 512, 512, 512, 512, and max pooling after 2, 4, 8, 12, 16 layers. After the 16 feature mapping layers, a final layer maps the features to 10 or 100 classes. In Table 1, SWSL(1L) to SWSL(5L) replace the first 1 to 5 layers with the rotation group convolutions with 4 orientations. We still keep the output feature maps of each layer the same between the baseline and SWSL, so that the replaced layers reduce the parameters and don't increase the amount of calculation. As shown in Table 1, SWSL(3L)(6.61%) on CIFAR10 outperforms the baseline (7.35%) with

Table 1: The results of VGG16. The RGIC contains 4 orientations.

Method	#Kernels	Err. (C10)	Err. (C100)
Baseline	64,64,128,128,256,256,256,256,512,...,512	7.35%	27.51%
SWSL(1L)	**16**,64,128,128,256,256,256,256,512,...,512	6.82%	27.56%
SWSL(2L)	**16,16**,128,128,256,256,256,256,512,...,512	6.79%	27.31%
SWSL(3L)	**16,16,32**,128,256,256,256,256,512,...,512	**6.61%**	26.82%
SWSL(4L)	**16,16,32,32**,256,256,256,256,512,...,512	7.02%	26.83%
SWSL(5L)	**16,16,32,32,64**,256,256,256,512,...,512	7.39%	**26.43%**

Table 2: The results of ResNet. SWSL(1L) replaces the first layer with rotation group convolution, and SWSL(1S) replace all layers in the first stage (with feature map size of 32×32). A 1×1 layer is inserted between the first and second stages of SWSL(1S) to mix the features in different orientations. For control, a 1×1 layer is also inserted in ResNet′. C10 and C100 represents CIFAR10 and CIFAR100, respectively.

Method	First layer Type	#Orient.	Err. (C10)	Err. (C100)
ResNet-20	normal	-	8.03%	32.85%
ResNet-20	repeat	-	8.96%	33.66%
SWSL(1L)	rotate	8	8.08%	32.67%
ResNet-32	normal	-	7.14%	30.85%
ResNet-32	repeat	-	7.63%	32.03%
SWSL(1L)	rotate	8	7.26%	30.81%
ResNet-110	normal	-	6.02%	27.31%
ResNet-110	repeat	-	6.43%	28.70%
SWSL(1L)	rotate	8	5.97%	27.48%

0.74% error lower. And on CIFAR100, SWSL(5L)(26.43%) outperforms the baseline (27.51%) with 1.08% error lower. SWSLs from 1L to 4L on CIFAR10 have lower errors (0.33% to 0.74% lower) than the baseline, and SWSL(5L) has slightly higher errors of 0.04%. Except that SWSL(1L) has slightly higher errors (0.05% higher), all SWSLs on CIFAR100 achieve better performances (0.20% to 1.08% lower errors).

For ResNet(1L), we only replace the initial layer before the residual blocks with rotation group convolution. The first layer of ResNet(1L) only has 2 kernels with 8 orientations, and it has 16 feature maps which are the same as the baseline. To study the effect on the number of feature maps, we also test on ResNet that the first layer only has 2 kernels and repeats 8 times to 16 feature maps. The results on Table 2 show that the SWSLs(1L) of ResNet-20/32/110 achieve similar errors compared with their baselines (range from -0.18% to 0.17%). And ResNet-20/32/110 with repeat feature maps, which have the same kernels and feature maps with SWSL(1L), show obvious error increases than both SWSL(1L) (range from 0.37% to 1.22%) and the baseline (range from 0.41% to 1.39%). It demonstrates that the benefits are from weight sharing instead of the number of feature maps.