NIPS 2017
LONG BEACH CA | DEC 4 - 9 | NIPS.CC

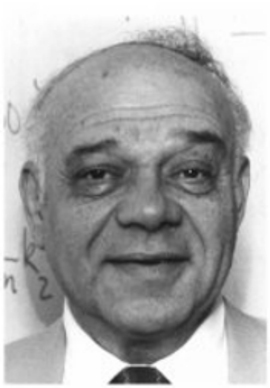
Deep Hyperspherical Learning

Weiyang Liu, Yan-Ming Zhang, Xingguo Li, Zhiding Yu, Bo Dai,
Tuo Zhao, Le Song

Georgia Institute of Technology

Motivation

- 2D Fourier Transform for images

A  The magnitude of A + The phase of B = 

B  The magnitude of B + The phase of A = 

- Phase contains the crucial discriminative information!

SphereNet: a network that focuses on the angular (phase) information

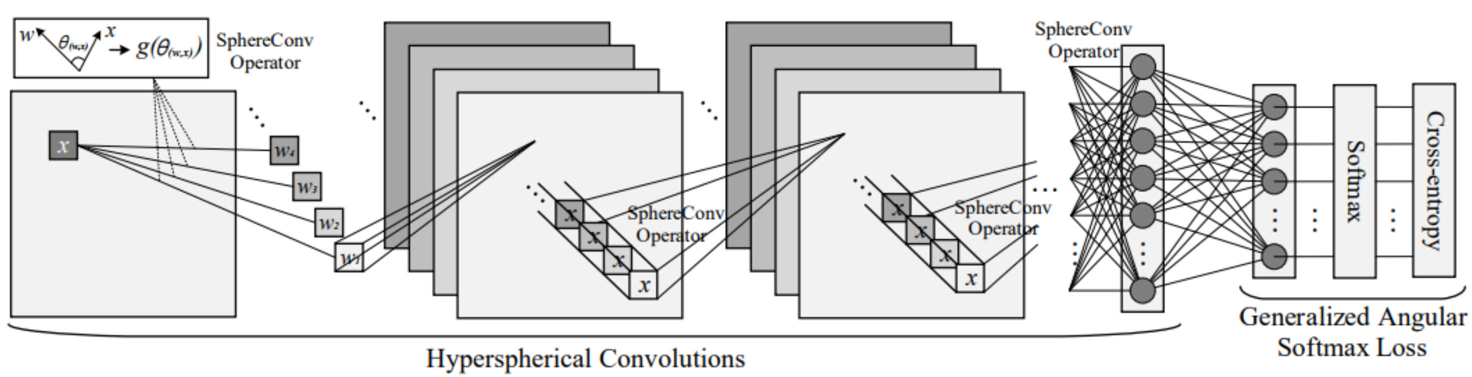
- Hyperspherical Convolutional (SphereConv) Operator:

$$\mathcal{F}_s(\boldsymbol{w}, \boldsymbol{x}) = g(\theta_{(\boldsymbol{w},\boldsymbol{x})}) + b_{\mathcal{F}_s}$$

  Where $\theta_{(\boldsymbol{w},\boldsymbol{x})}$ is the angle between the kernel parameter w and the local patch x. A simple example is cosine SphereConv:

$$g(\theta_{(\boldsymbol{w},\boldsymbol{x})}) = \cos(\theta_{(\boldsymbol{w},\boldsymbol{x})})$$

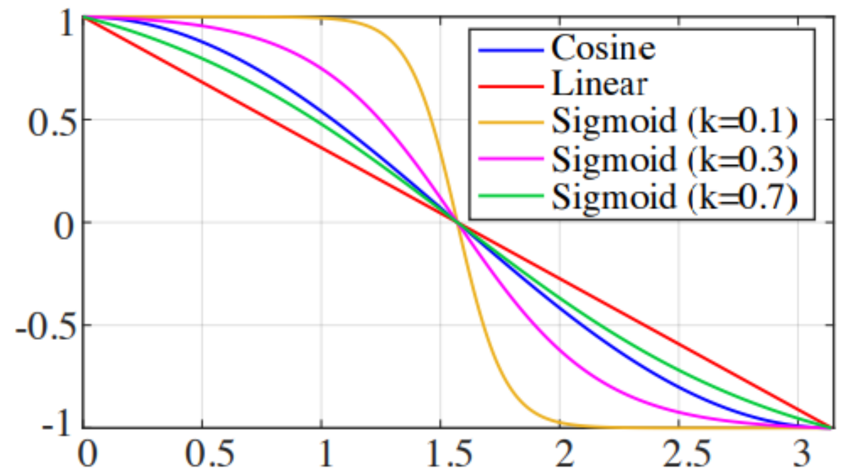
- We use this SphereConv operator to replace the original inner product based convolutional operator in the CNNs, and propose the *SphereNet*. (SphereNet comes from that angle can be viewed as the geodesic distance on a unit hypersphere)



Four SphereConv operators

- *linear SphereConv*

$$g(\theta_{(\boldsymbol{w},\boldsymbol{x})}) = a\theta_{(\boldsymbol{w},\boldsymbol{x})} + b$$



- *cosine SphereConv*

$$g(\theta_{(\boldsymbol{w},\boldsymbol{x})}) = \cos(\theta_{(\boldsymbol{w},\boldsymbol{x})})$$

- *sigmoid SphereConv*

$$g(\theta_{(\boldsymbol{w},\boldsymbol{x})}) = \frac{1+\exp(-\frac{\pi}{2k})}{1-\exp(-\frac{\pi}{2k})} \cdot \frac{1-\exp\left(\frac{\theta_{(\boldsymbol{w},\boldsymbol{x})}}{k} - \frac{\pi}{2k}\right)}{1+\exp\left(\frac{\theta_{(\boldsymbol{w},\boldsymbol{x})}}{k} - \frac{\pi}{2k}\right)}$$

- *Learnable SphereConv*

$$g(\theta_{(\boldsymbol{w},\boldsymbol{x})}) = \frac{1+\exp(-\frac{\pi}{2k})}{1-\exp(-\frac{\pi}{2k})} \cdot \frac{1-\exp\left(\frac{\theta_{(\boldsymbol{w},\boldsymbol{x})}}{k} - \frac{\pi}{2k}\right)}{1+\exp\left(\frac{\theta_{(\boldsymbol{w},\boldsymbol{x})}}{k} - \frac{\pi}{2k}\right)}$$

with the parameter k to be learned in back-prop

Theoretical Insights

- Suppose the observation is $\boldsymbol{F} = \boldsymbol{U}^*\boldsymbol{V}^{*\top}$ (ignore the bias), where $\boldsymbol{U}^* \in \mathbb{R}^{n\times k}$ is the weight, $\boldsymbol{V}^* \in \mathbb{R}^{m\times k}$ is the input that embeds weights from previous layers.

Scaling issue of neural networks:

- Consider the objective: $\min_{\boldsymbol{U}\in\mathbb{R}^{n\times k},\boldsymbol{V}\in\mathbb{R}^{m\times k}} \mathcal{G}(\boldsymbol{U},\boldsymbol{V}) = \frac{1}{2}\|\boldsymbol{F} - \boldsymbol{U}\boldsymbol{V}^\top\|_{\mathrm{F}}^2$
- *Lemma1: Consider a pair of global optimal points* $\boldsymbol{U},\boldsymbol{V}$ *satisfying* $\boldsymbol{F} = \boldsymbol{U}\boldsymbol{V}^\top$ *and* $\mathrm{Tr}(\boldsymbol{V}^\top\boldsymbol{V}\otimes\boldsymbol{I}_n) \leq \mathrm{Tr}(\boldsymbol{U}^\top\boldsymbol{U}\otimes\boldsymbol{I}_m)$. *For any real* $c > 1$, *let* $\widetilde{\boldsymbol{U}} = c\boldsymbol{U}$ *and* $\widetilde{\boldsymbol{V}} = \boldsymbol{V}/c$, *then we have* $\kappa(\nabla^2\mathcal{G}(\widetilde{\boldsymbol{U}},\widetilde{\boldsymbol{V}})) = \Omega(c^2\kappa(\nabla^2\mathcal{G}(\boldsymbol{U},\boldsymbol{V})))$, *where* $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$ *is the restricted condition number with* $\lambda_{\max}$ *being the largest and* $\lambda_{\min}$ *being the smallest nonzero eigenvalues.*

Insensitiveness to Scaling for SphereConv:

- Consider our proposed cosine SphereConv operator, an equivalent problem is:

$$\min_{\boldsymbol{U}\in\mathbb{R}^{n\times k},\boldsymbol{V}\in\mathbb{R}^{m\times k}} \mathcal{G}_S(\boldsymbol{U},\boldsymbol{V}) = \frac{1}{2}\|\boldsymbol{F} - \boldsymbol{D}_{\boldsymbol{U}}\boldsymbol{U}\boldsymbol{V}^\top\boldsymbol{D}_{\boldsymbol{V}}\|_{\mathrm{F}}^2$$

  where $\boldsymbol{D}_{\boldsymbol{U}} = \mathrm{diag}(\frac{1}{\|\boldsymbol{U}_{1,:}\|_2},\dots,\frac{1}{\|\boldsymbol{U}_{n,:}\|_2}) \in \mathbb{R}^{n\times n}$ and $\boldsymbol{D}_{\boldsymbol{V}} = \mathrm{diag}(\frac{1}{\|\boldsymbol{V}_{1,:}\|_2},\dots,\frac{1}{\|\boldsymbol{V}_{m,:}\|_2}) \in \mathbb{R}^{m\times m}$ are diagonal matrices.

- *Lemma2: For any real* $c > 1$, *let* $\widetilde{\boldsymbol{U}} = c\boldsymbol{U}$ *and* $\widetilde{\boldsymbol{V}} = \boldsymbol{V}/c$, *then we have* $\lambda_i(\nabla^2\mathcal{G}_S(\widetilde{\boldsymbol{U}},\widetilde{\boldsymbol{V}})) = \lambda_i(\nabla^2\mathcal{G}_S(\boldsymbol{U},\boldsymbol{V}))$ *for all* $i \in [(n+m)k] = \{1,2,\dots,(n+m)k\}$ *and* $\kappa(\nabla^2\mathcal{G}(\widetilde{\boldsymbol{U}},\widetilde{\boldsymbol{V}})) = \kappa(\nabla^2\mathcal{G}(\boldsymbol{U},\boldsymbol{V}))$. *, where* κ *is defined as in Lemma1.*

- Regular Neural Nets: scales as $\Omega(c^2)$
- SphereConv: ***<u>insensitive</u>*** to scaling

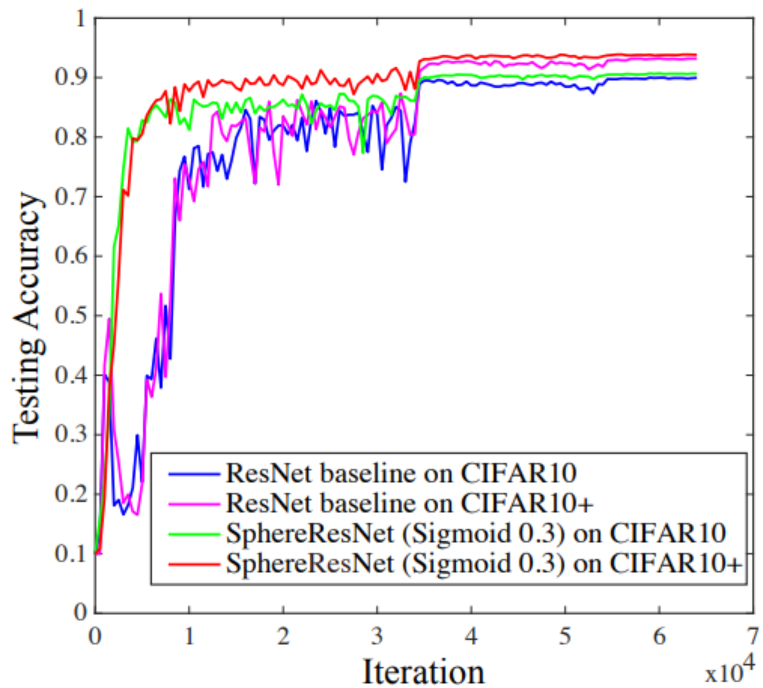
More on SphereNets

- SphereConv can also be used to the fully connected layers, recurrent layers, etc.
- SphereConv can also be viewed as a normalization method that could avoid covariate shift (due to the bounded outputs), and can work simultaneously with Batch Normalization.
- We also design angular loss functions for *SphereConv*, i.e., generalized angular softmax (GA-Softmax) loss

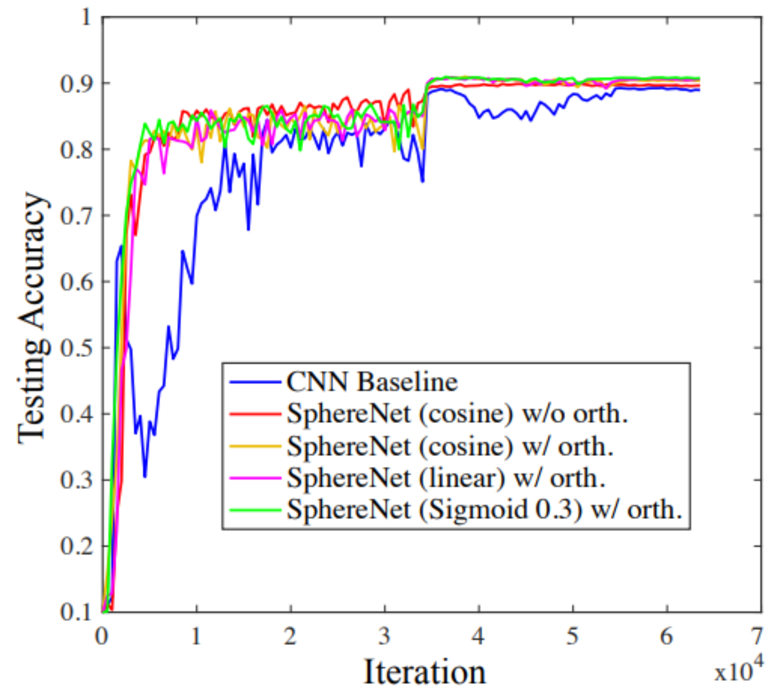
$$L_i = -\log\left(\frac{e^{\|\boldsymbol{x}_i\| g(m\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\| g(m\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|\boldsymbol{x}_i\| g(\theta_{j,i})}}\right)$$

Experiments

- Faster Convergence and better accuracy on CIFAR-10, CIFAR-100



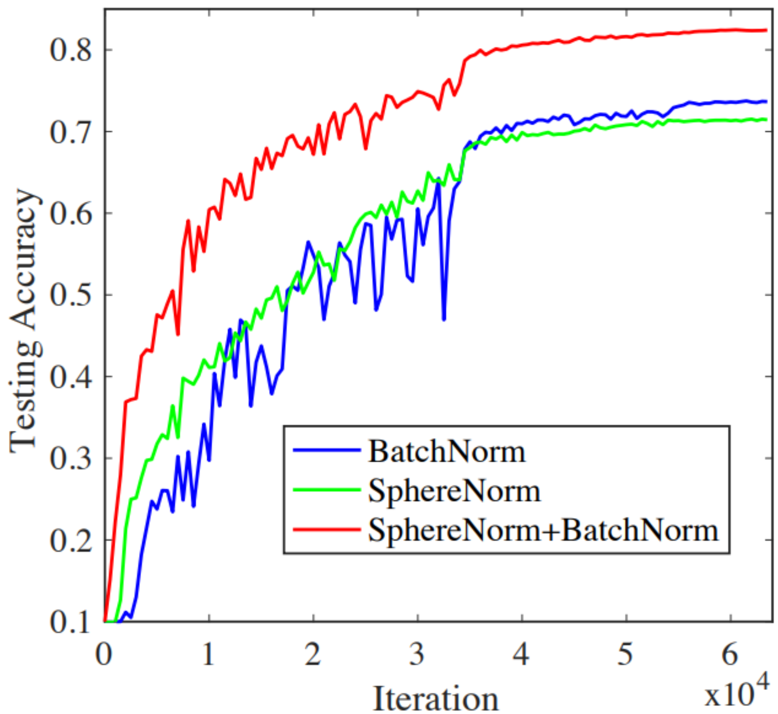
(a) ResNet vs. SphereResNet on CIFAR-10/10+



(b) CNN vs. SphereNet (orth.) on CIFAR-10

Experiments

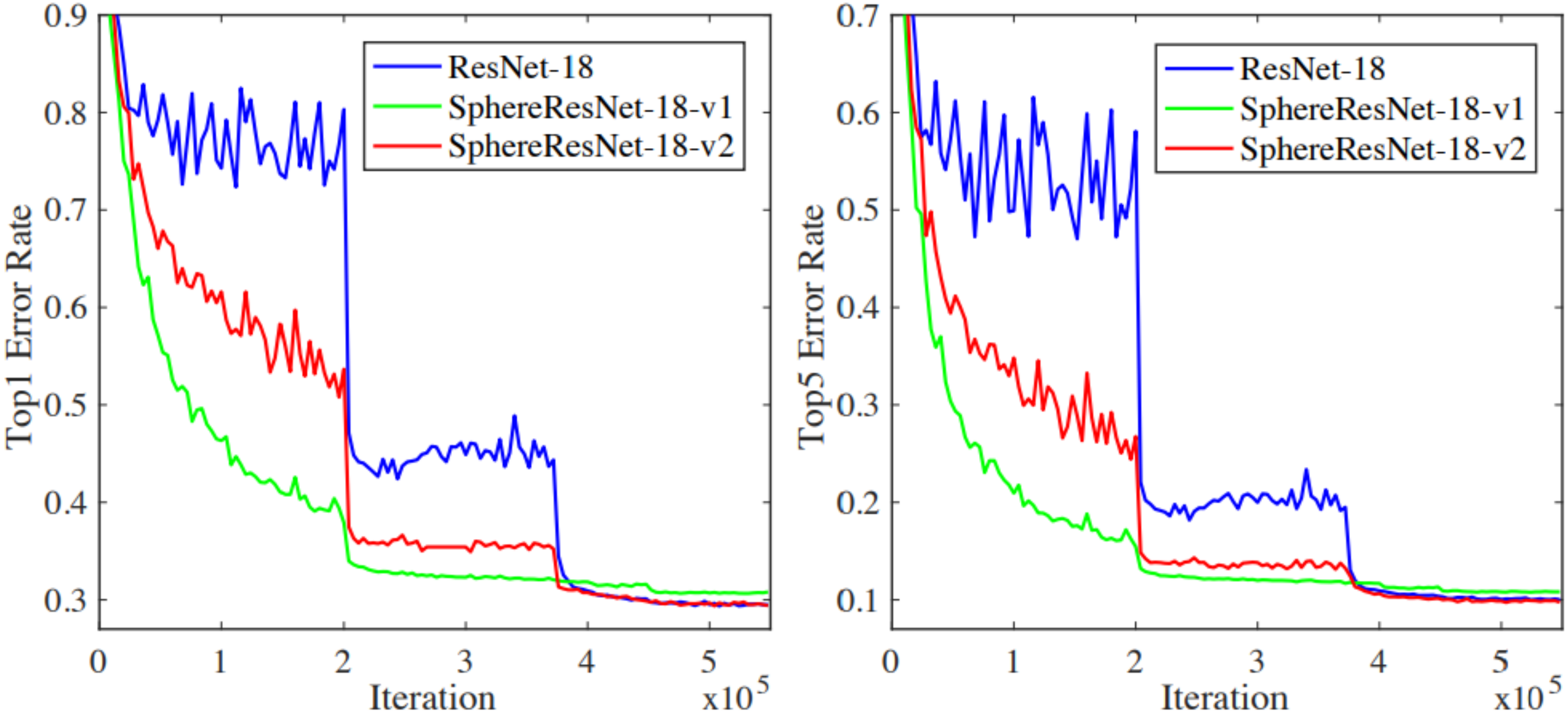
- SphereConv can be used as a new normalization method (SphereNorm), comparable to Batch Normalization. But they can be used simultaneously.
- The advantages of SphereNorm are very significant, especially with small mini-batch size.



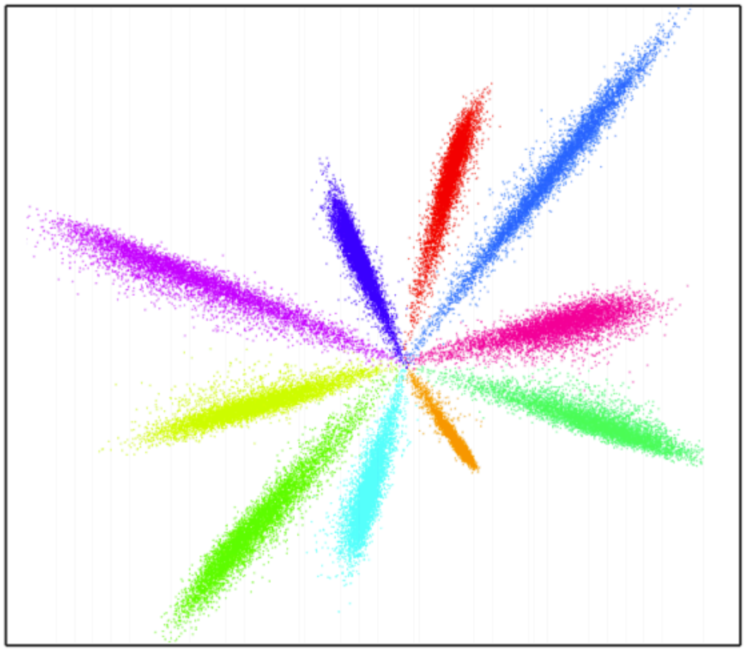
Mini-batch size =4!

Experiments

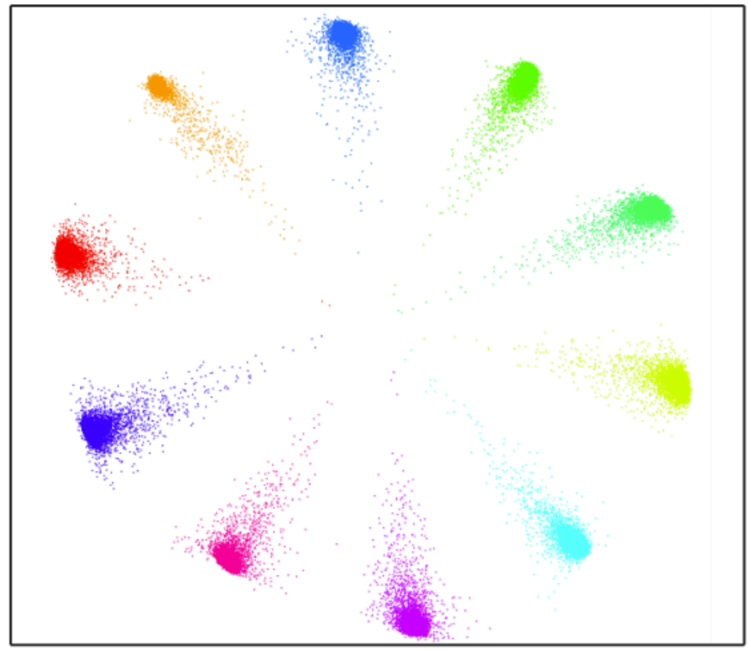
- Faster Convergence and comparable accuracy on Imagenet-2012



Visualization on MNIST



Original CNN



SphereNet

More experiments

- Using the SphereConv only to the last fully connected layer gives impressive results on face recognition.

Method	protocol	Rank1 Acc.	Ver.
NTechLAB - facenx large	Large	73.300	85.081
Vocord - DeepVo1	Large	**75.127**	67.318
Deepsense - Large	Large	74.799	**87.764**
Shanghai Tech	Large	74.049	86.369
Google - FaceNet v8	Large	70.496	86.473
Beijing FaceAll_Norm_1600	Large	64.804	67.118
Beijing FaceAll_1600	Large	63.977	63.960
Deepsense - Small	Small	**70.983**	**82.851**
SIAT_MMLAB	Small	65.233	76.720
Barebones FR - cnn	Small	59.363	59.036
NTechLAB - facenx_small	Small	58.218	66.366
3DiVi Company - tdvm6	Small	33.705	36.927
Softmax Loss	Small	54.855	65.925
Softmax+Contrastive Loss [26]	Small	65.219	78.865
Triplet Loss [22]	Small	64.797	78.322
L-Softmax Loss [16]	Small	67.128	80.423
Softmax+Center Loss [34]	Small	65.494	80.146
SphereFace (single model)	Small	**72.729**	**85.561**
SphereFace (3-patch ensemble)	Small	**75.766**	**89.142**

The End

- The code will be made available at

https://github.com/wy1iu/SphereNet

- The code of SphereFace is available at

https://github.com/wy1iu/sphereface