VarGFaceNet: An Efficient Variable Group Convolutional Neural Network for Lightweight Face Recognition

Mengjia Yan
Horizon Robotics
mengjia.yan@horizon.ai

Mengao Zhao
Horizon Robotics
mengao.zhao@horizon.ai

Zining Xu
Horizon Robotics
zining.xu@horizon.ai

Qian Zhang
Horizon Robotics
qian01.zhang@horizon.ai

Guoli Wang
Horizon Robotics
guoli.wang@horizon.ai

Zhizhong Su
Horizon Robotics
zhizhong.su@horizon.ai

Abstract

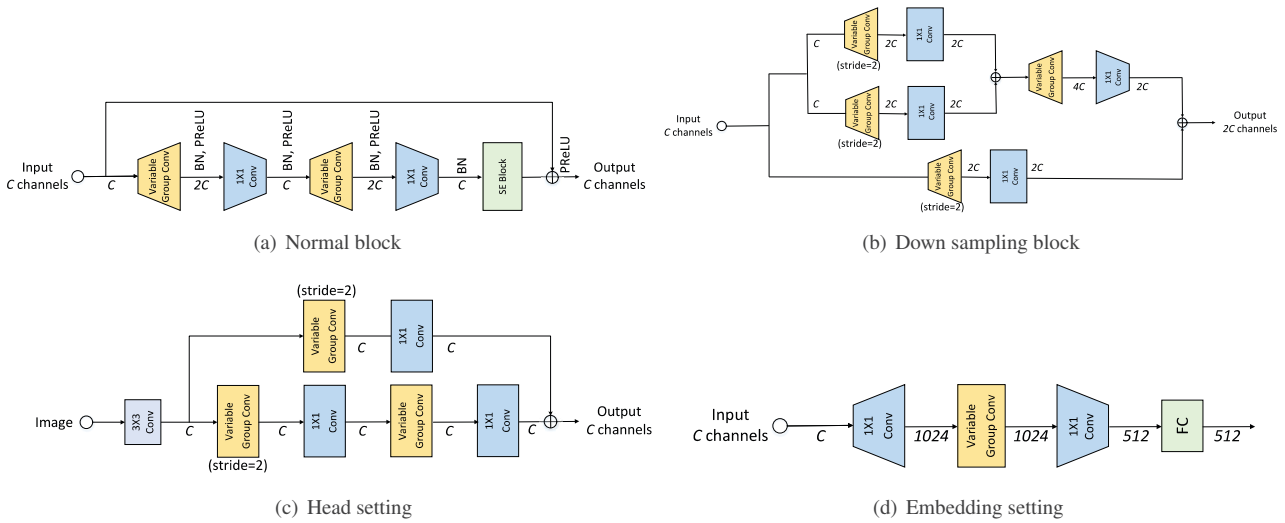
To improve the discriminative and generalization ability of lightweight network for face recognition, we propose an efficient variable group convolutional network called VarGFaceNet. Variable group convolution is introduced by VarGNet to solve the conflict between small computational cost and the unbalance of computational intensity inside a block. We employ variable group convolution to design our network which can support large scale face identification while reduce computational cost and parameters. Specifically, we use a head setting to reserve essential information at the start of the network and propose a particular embedding setting to reduce parameters of fully-connected layer for embedding. To enhance interpretation ability, we employ an equivalence of angular distillation loss to guide our lightweight network and we apply recursive knowledge distillation to relieve the discrepancy between the teacher model and the student model. The champion of deepglint-light track of LFR (2019) challenge demonstrates the effectiveness of our model and approach. Implementation of VarGFaceNet will be released at https://github.com/zma-c-137/VarGFaceNet soon.

1. Introduction

With the surge of computational resources, face recognition using deep representation has been widely applied to many fields such as surveillance, marketing and biometrics[3, 17]. However, it is still a challenging task to implement face recognition on limited computational cost system such as mobile and embedded systems because of the large scale identities needed to be classified.

Many work propose lightweight networks for common computer vision tasks such as *SqueezeNet*[15], *MobileNet* [12], *MobileNetV2* [20], *ShuffleNet* [26]. *SqueezeNet*[15] extensively uses 1×1 convolution, achieving $50\times$ fewer parameters than *AlexNet*[16] while maintains AlexNet-level accuracy on ImageNet. *MobileNet*[12] utilizes depthwise separable convolution to achieve a trade off between latency and accuracy. Based on this work, *MobileNetV2*[20] proposes an inverted bottleneck structure to enhance discriminative ability of network. *ShuffleNet*[26] uses pointwise group convolution and channel shuffle operations to further reduce computation cost. Even though they cost small computation during inference and achieve good performance on various applications, optimization problems on embedded system still remain on embedded hardware and corresponding compilers [25]. To handle this conflict, *VarGNet* [25] proposes a variable group convolution which can efficiently solve the unbalance of computational intensity inside a block. Meanwhile, we explore that variable group convolution has larger capacity than depthwise convolution with the same kernel size, which helps network to extract more essential information. However, *VarGNet* is designed for general tasks such as image classificaiton and object detection. It decreases spatial area to the half in the head setting to save memory and computational cost, while this setting is not suitable for face recognition task since detailed information of face is necessary. And there is only an average pooling layer between last conv and fully connected layer of the embedding, which may not extract enough discriminative information.

Based on *VarGNet*, we propose an efficient variable group convolutional network for lightweight face recognition, shorted as VarGFaceNet. In order to enhance the discriminative ability of *VarGNet* for large scale face recognition task, we first add SE block [13] and PReLU [8] on blocks of *VarGNet*. Then we remove the downsample process at the start of network to preserve more information. To decrease parameters of network, we apply variable group



(a) Normal block

(b) Down sampling block

(c) Head setting

(d) Embedding setting

Figure 1. Settings of VarGFaceNet. a) is the normal block of VarGFaceNet. We add SE block on normal block of VarGNet. b) is the down sampling block. c) is head setting of VarGFaceNet. We do not use downsample in first conv in order to keep enough information. c) is the embedding setting of VarGFaceNet. We first expand channels from 320 to 1024. Then we employ variable group convolution and pointwise convolution to reduce the parameters and computational cost while remain essential information.

convolution to shrink the feature tensor to $1 \times 1 \times 512$ before fc layer. The performance of VarGFaceNet demonstrates that this embedding setting can preserve discriminative ability while reduce parameters of the network.

To enhance the interpretation ability of lightweight network, we apply knowledge distillation during the training. There are several approaches aim at making the deep network smaller and cost-efficient, such as model pruning, model quantization and knowledge distillation. Among them, knowledge distillation is being actively investigated due to its architectural flexibility. Hinton[11] introduces the concept of knowledge distillation and proposes to use the softmax output of teacher network to achieve knowledge distillation. To take better advantage of information from teacher network, FitNets[19] adopts the idea of feature distillation and encourages student network to mimic the hidden feature values of teacher network. After FitNets, there are variant methods attempt to exploit the knowledge of teacher network, such as transferring the feature activation map[10], activation-based and gradient-based Attention Maps[24]. Recently *ShrinkTeaNet* [6] introduces an angular distillation loss to focus on angular information of teacher model. Inspired by angular distillation loss we employ an equivalent loss with better implementation efficiency as the guide of VarGFaceNet. Moreover, to relieve the complexity of optimization caused by the discrepancy between teacher model and student model, we introduce recursive knowledge distillation which treats the model of student trained in one generation as pretrained model for the next generation.

We evaluate our model and approach on LFR challenge [4]. LFR challenge is a lightweight face recognition challenge which requires networks whose FLOPs is under 1G and memory footprint is under 20M. VarGFaceNet achieves the state-of-the-art performance on this challenge which is shown in Section 3. Our contributions are summarized as follows:

- To improve the discriminative ability of *VarGNet* [25] in large-scale face recognition we employ a different head setting and propose a new embedding block. In embedding block, we first expand channels to 1024 by 1×1 convolution layer to reserve essential information, then we use variable group conv and pointwise conv to shrink the spatial area to 1×1 while saving computational cost. These settings improve the performance on face recognition tasks which shown in Section 3.

- To imporve the generalization ability of lightweight models, we propose recursive knowledge distillation which relieves the generalization gap between teacher models and student models in one generation.

- We analyse the efficiency of variable group convolution and employ an equivalence of angular distillation loss during training. Experiments conducted to show the effectiveness of our approach.

2. Approach

2.1. Variable Group Convolution

Group Convolution was first introduced in *AlexNet* [16] for computational cost reduction on GPUs. Then, the cardinality of group convolution demonstrated a better performance than the dimensions of depth and width in *ResNext* [22]. Designed for mobile device, *MobileNet* [12] and *MobileNetV2* [20] proposed depthwise separable convolution inspired by group convolution to save computational cost while keep discriminative ability of convolution. However, depthwise separable convolution spends 95% computation time in Conv 1×1, which causes a large MAdds gap between two consecutive laysers (Conv 1×1 and Conv DW 3×3) [12]. This gap is unfriendly to embedded systems who load all weights of the network to perform convolution[23]: embedded systems need extra buffers for Conv 1×1.

To keep the balance of computational intensity inside a block, *VarGNet* [25] sets the channel numbers in a group as a constant S. The constant channel numbers in a group lead to the variable number of groups n_i in a convolution, named variable group convolution. The computational cost of a variable group convolution is:

$$k^2 \times h_i \times w_i \times S \times c_{i+1} \tag{1}$$

$$S = \frac{c_i}{n_i} \tag{2}$$

The input of this layer is $h_i \times w_i \times c_i$ and the output of that is $h_i \times w_i \times c_{i+1}$. k is the kernel size. When variable group convolution is used to replace depthwise convolution in *MobileNet* [12], the computational cost of pointwise convolution is:

$$1^2 \times h_i \times w_i \times c_{i+1} \times c_{i+2} \tag{3}$$

The ratio of computational cost between variable group convolution and pointwise convolution is $\frac{k^2 S}{c_{i+2}}$ while that between depthwise convolution and pointwise convolution is $\frac{k^2}{c_{i+2}}$. In practice, $c_{i+2} \gg k^2$, $S > 1$, so $\frac{k^2 S}{c_{i+2}} > \frac{k^2}{c_{i+2}}$. Hence, it will be more computational balanced inside a block when employs variable group convolution on the bottom of pointwise convolution instead of depthwise convolution.

Moreover, $S > 1$ means variable group convolution has higher MAdds and larger network capacity than depthwise convoluiton (with the same kernel size), which is capable of extracting more information.

2.2. Blocks of Variable Group Network

Communication between off-chip memory and on-chip memory only happens on the start and the end of block computing when a block is grouped and computed together on embedded systems [23]. To limit the communication cost, *VarGNet* sets the number of output channels to be same as the number of input channels in the normal block. Meanwhile, *VarGNet* expands the C channels at the start of the block to $2C$ channels using variable group convolution to keep the generalization ability of the block. The normal block we used is shown in Fig. 1(a), and down sampling block is shown in Fig. 1(b). Different from the blocks in *VarGNet* [25], we add SE block in normal block and employ PReLU instead of ReLU to increase the discriminative ability of the block.

2.3. Lightweight Network for Face Recognition

2.3.1 Head setting

The main challenge of face recognition is the large scale identities involved in testing/training phase. It requires discriminative ability as much as possible to support distinguishing millions of identities. In order to reserve this ability in lightweight networks, we use 3×3 Conv with stride 1 at the start of network instead of 3×3 Conv with stride 2 in *VarGNet*. It is similar to the input setting of [3]. The output feature size of first conv in *VarGNet* will be downsampled while ours remains the same as input size, shown in Fig. 1(c).

2.3.2 Embedding setting

To obtain the embedding of faces, many work [3, 17] employ a fully-connected layer directly on the top of last convolution. However, the parameters of this fully-connected layer will be huge when output features from last convoluiton are relatively large. For instance, in ResNet 100 [3] the output of last conv is $7 \times 7 \times 512$, and the parameters of fc layer (embedding size is 512) are $7 \times 7 \times 512 \times 512$. The overall parameters of fc layer for embedding are 12.25M, and the memory footprint is 49M (float32)!

In order to design a lightweight network (memory footprint is less than 20M, FLOPs is less than 1G), we employ variable group convolution after last conv to shrink the feature maps to $1 \times 1 \times 512$ before fc layer. Consequently, the memory footprint of fc layer for embedding is only 1M. Fig.1(d) shows the setting of embedding block. Shrinking the feature tensor to $1 \times 1 \times 512$ before fc layer for embedding is risky since information contains by this feature tensor is limited. To avoid the derease of essential information, we expand channels after last conv to retain as much information as possible. Then we employ variable group convolution and pointwise convolution to decrease the parameters and computational cost while keep information.

Specifically, we first use a 1×1 Conv to expand the channels from 320 to 1024. Then we employ a 7×7 variable group convolution layer (8 channels in a group) to shrink the feature tensors from $7 \times 7 \times 1024$ to $1 \times 1 \times 1024$. Finally,

Layer	Output Size	KSize	Stride	Repeat	Output Channels
Image	112x112				3
Conv 1	112x112	3x3	1	1	40
Head Block	56x56		2	1	40
Stage2	28x28		2	1	80
	28x28		1	2	
Stage3	14x14		2	1	160
	14x14		1	6	
Stage4	7x7		2	1	320
	7x7		1	3	
Conv 5	7x7	1x1	1	1	1024
Group Conv	1x1	7x7	1	1	1024
Pointwise Conv	1x1	1x1	1	1	512
FC					512

Table 1. Overall architecture of VarGFaceNet. It only has 1G FLOPs and 5M parameters (memory footprint is 20M saved as float32).

pointwise convolution is used to connect the channels and output the feature tensors to $1 \times 1 \times 512$. The new embedding block setting only takes up 5.78M while the original fc layer takes up 30M ($7 \times 7 \times 320 \times 512$) on the disk.

Experiments of comparison between our network and *VarGNet* in Section 3.3 demonstrate the efficiency of our network on face recognition tasks.

2.3.3 Overall architecture

The overall architecture of our lightweight network (VarGFaceNet) is illustrated in Table 1. The memory footprint of our VarGFaceNet is 20M and FLOPs is 1G. We set $S = 8$ in a group empirically. Benefit from variable group convolution, head settings and particular embedding settings, VarGFaceNet can achieve good performance on face recognition task with limited computational cost and parameters. In Section 3, we will demonstrate the effectiveness of our network on a million distractors face recognition task.

2.4. Angular Distillation Loss

Knowledge distillation has been widely used in lightweight network training since it can transfer the interpretation ability of a big network to a smaller network [12]. Majority tasks that used knowledge distillation are close set tasks [19, 11]. They apply scores/logits or embeddings/feature magnitude to compute $l2$ distance or cross entropy as loss. However, for open set tasks, scores/logits of training set contain limited information of testing set and the exact match of featuers maybe over-regularized in some situations. To extract useful information and avoid over-regularization, [6] proposes an angular distillation loss for knowledge distillation:

$$L_a(F_t^i, F_s^i) = -\frac{1}{m}\sum_{i=1}^{m} ||1 - \frac{F_t^i}{||F_t^i||} * \frac{F_s^i}{||F_s^i||}||_2^2 \quad (4)$$

F_t^i is the ith feature of teacher model, F_s^i is ith features of student model. m is the number of samples in a batch. Eq. 4 first computes cosine similarity between features of teacher and student, then minimizes the $l2$ distance between this similarity and 1. Inspired by [6], we propose to use Eq. 5 to enhance the implementation efficiency. Since cosine similarity is less than 1, minimize Eq. 4 is equivalent to minimize Eq. 5.

$$L_s(F_t^i, F_s^i) = -\frac{1}{m}\sum_{i=1}^{m} ||\frac{F_t^i}{||F_t^i||} - \frac{F_s^i}{||F_s^i||}||_2^2 \quad (5)$$

Compared with previous $l2$ loss of exact features, Eq. 4 and Eq. 5 focus on angular information and the distribution of embeddings.

In addition, we employ arcface [3] as our classification loss which also pays attention to angular information:

$$L_{Arc} = -\frac{1}{m}\sum_{i=1}^{m} \log \frac{e^{s(cos(\Theta_{y_i}+m))}}{e^{s(cos(\Theta_{y_i}+m))} + \sum_{j=1,j\neq y_i}^{n} e^{scos\Theta_j}} \quad (6)$$

To sum up, the objective function we used in training is:

$$L = L_{Arc} + \alpha L_s \quad (7)$$

We empirically set $\alpha = 7$ in our implementation.

2.5. Recursive Knowledge Distillation

Knowledge distillation with one generation is sometimes difficult to transfer enough knowledge when large discrepancy exists between teacher models and student models.

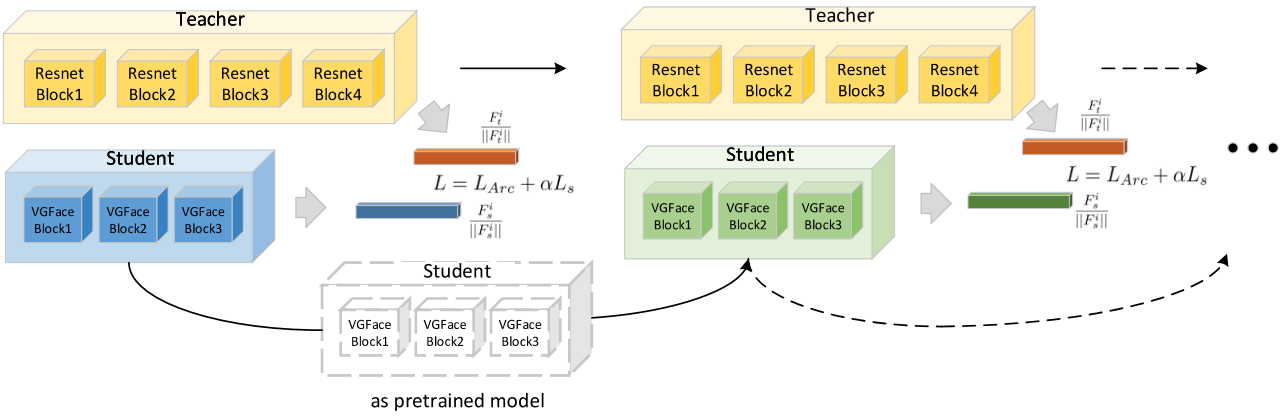


Figure 2. The process of recursive knowledge distillation. We apply the first generation of student to initialize the second generation of student while the teacher model is remained. Angular distillation loss and arcface loss are used to guide training.

Network	LFW	CFP-FP	AgeDB-30	deepglint-light (TPR@FPR=1e-8)	Flops
y2	0.99700	0.97829	0.97517	0.803	933M
VarGFaceNet	0.99683	0.98086	0.98100	0.855	1022M

Table 2. VarGFaceNet vs. y2. Performance is recorded within the same epoch. The validation performance of VarGFaceNet is 0.6% and 0.2% higher than y2 on AgeDB-30 and CFP-FP respectively. Testing result of VarGFaceNet is 5% higher than y2.

For instance, in our implementation, the FLOPs of teacher model is 24G while that of student model is 1G. And the number of parameters of teacher model is 108M while that of student model is 5M. Moreover, the different architecture and block settings between teacher model and student model increase the complexity of training as well. To improve the discriminative and generalization ability of our student network, we propose recursive knowledge distillation, which employs the first generation of student to initialize the second generation of student, as shown in Fig. 2.

In recursive knowledge distillation, we employ the same teacher model in all generations. That means the angular information of samples which guides the student model is invariable. There are two merits if we use recursive knowledge distillation:

1 It will be easier to approach guided direction of teacher when apply a good initialization.

2 The conflicts between margin of classification loss and guided angular information in the first generation will be relieved in the next generation.

The results of our experiments in Section 3 illustrate the performance of recursive knowledge distillation.

In this section, we first introduce the datasets and evaluation metric. Then, to demonstrate the effectiveness of our VarGFaceNet, we compare our network with y2 network(a deeper mobilefacenet[2, 3]). After that, the investigation for the effect of different teacher models in knowledge distillation is revealed. Finally, we show the competitive performance of VarGFaceNet using recursive knowledge distillation on LFR2019 Challenge.

3. Experiments

3.1. Datasets and Evaluation Metric

We employ the dataset(clean from MS1M[7]) provided by LFR2019 for training. All face images in this dataset are aligned by five facial landmarks predicted from RetinaFace[5] then resized to 112×112. There are 5.1M images collected from 93K identities. For test set, Trillion-pairs dataset [1] is used. It contains two parts: 1) ELFW: Face images of celebrities in the LFW name list. There are 274K images from 5.7K identities; 2) DELFW: Distractors for ELFW. There are 1.58 M face images from Flickr. All test images are preprocessed and resized to 112×112. We refer deepglint-light to trillionpairs testing set in the following. During the training, we utilize face verification datasets (e.g. LFW[14], CFP-FP[21], AgeDB-30[18]) to validate different settings using 1:1 verification protocol. Moreover, we employ the TPR@FPR=1e-8 as evaluation metric for identification.

Method	LFW	CFP-FP	AgeDB-30	deepglint-light (TPR@FPR=1e-8)
teacher	0.99683	0.98414	0.98083	0.86846
student	0.99683	0.98171	0.97550	0.84341
teacher	0.99817	0.98729	0.98133	0.90231
student	0.99733	0.98200	0.98100	0.85461
teacher	0.99833	0.99057	0.98250	0.93315
student	0.99783	0.98400	0.98067	0.88334

Table 3. Performance of VarGFaceNet with the guide of different teacher models. Performance is recorded within the same epoch. Results of CFP-FP(validation set) and deepglint-light(TPR@FPR=1e-8) (testing set) show that the higher performance of teacher model leads to the better results of student model.

Network	LFW	CFP-FP	AgeDB-30	Flops
r100(teacher)	0.9987	0.9917	0.9852	24G
VarGNet(student)	0.9977	0.9810	0.9810	1029M
VarGFaceNet(student)	0.9985	0.9850	0.9815	1022M

Table 4. VarGFaceNet vs. VarGNet. We show the highest performance of every validation dataset. The performance of VarGFaceNet is higher than VarGNet on LFW, AgeDB-30 and CFP-FP.

3.2. VarGFaceNet train from scratch

To validate the efficiency and effectiveness of VarGFaceNet, we first train our network from scratch, and compare the performance with mobilefacenet(y2) [2, 3]. We employ arcface loss as the objective function of classification during training. Tabel 2 presents the comparison results of VarGFaceNet and y2. It can be observed that under the limitation of 1G FLOPs, VarGFaceNet is able to reach better face recognition performance on validation sets. Compared with y2, our verification results of AgeDB-30 , CFP-FP have increased 0.6% and 0.2% respectively, testing result of deepglint-light (TPR@FPR=1e-8) has increased 5%. There are two intuitions for the better performance: 1. our network can contain more parameters than y2 when limit FLOPs because of variable group convolution. The biggest number of channels is 256 in y2 while ours is 320 before last conv. 2. Our embedding setting can extract more essential information. y2 expands the number of channels from 256 to 512 then use 7×7 depthwise convolution to get the feature tensor before fc layer. We expand the number of channels from 320 to 1024 then use variable group convolution and pointwise convolution which have larger network capacity.

3.3. VarGFaceNet guided by ResNet

In order to achieve higher performance than train from scratch, bigger networks are applied to perform knowledge distillation using angular distillation loss. Moreover, we conduct experiments to investigate the effect of different teacher models on VarGFaceNet. We employ ResNet 100 [9] with SE as our teacher model. The teacher model has 24G FLOPs and 108M parameters. The results are illustrated in Tabel 3. It can be observed that 1. even though the architectures of teacher and student are quite different, VarGFaceNet still approaches the performance of ResNet; 2. the performance of VarGFaceNet is highly correlated with teacher model. The higher performance teacher model has, the better interpretation ability VarGFaceNet will learn.

To validate the efficiency of our settings, we conduct comparison experiments between our network and *VarGNet*. Using the same teacher network, we change the head setting of *VarGNet* to our head setting for fair comparison and use the same loss function as above. In Tabel 4, the plain *VarGNet* has lower accuracy in LFW, CFP-FP, AgeDB-30. There is only an average pooling between last conv and fc layer in *VarGNet*. The results illustrate that our embedding setting is more suitable for face recognition task since it can extract more essential information.

3.4. Recursive Knowledge Distillation

As we discuss in Section 2.5, when there is a large discrepancy between teacher model and student, knowledge distillation for one generation may not enough for knowledge transfer. To validate it, we use ResNet 100 model as our teacher model, and conduct recursive knowledge distillation on VarGFaceNet. A performance improvement shown in Table 5 when we train the model in next generation. The varification result of LFW and CFP-FP is increased by 0.1% while testing result of deepglint-light(TPR@FPR=1e-8) is 0.4% higher than pervious generation. Furthermore, we believe that it will lead to better performance if we continue to conduct training in more generations.

Method	LFW	CFP-FP	AgeDB-30	deepglint-light (TPR@FPR=1e-8)
recursive=1	0.99783	0.98400	0.98067	0.88334
recursive=2	0.99833	0.98271	0.98050	0.88784

Table 5. Performance of recursive knowledge distillation. Performance is recorded within the same epoch.

Verification results of LFW, AgeDB-30 are increased in the second generation. Performance of testing set deepglint-light(TPR@FPR=1e-8) is increased by 0.4% the same time.

4. Conclusion

In this paper, we propose an efficient lightweight network called VarGFaceNet for large scale face recognition. Benefit from variable group convolution, VarGFaceNet is capable of finding a better trade-off between efficiency and performance. The head setting and embedding setting specific to face recogniton help preserve information while reduce parametes. Moreover, to improve the interpretation ability of lightweight network, we employ an equivalence of angular distillation loss as our objective function and present a recursive knowledge distillation strategy. The state-of-the-art performance on LFR challenge demonstrates the superiority of our method.

Acknowledgments We would like to thank Xin Wang, Helong Zhou, Zhichao Li, Xiao Jiang, Yuxiang Tuo for their helpful discussion.

References

[1] Trillionpairs. `http://trillionpairs.deepglint.com/overview`. Accessed July, 2019.

[2] S. Chen, Y. Liu, X. Gao, and Z. Han. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In *Chinese Conference on Biometric Recognition*, pages 428–438. Springer, 2018.

[3] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[4] J. Deng, J. Guo, D. Zhang, Y. Deng, X. Lu, S. Shi, and S. Zafeiriou. Lightweight face recognition challenge. *In Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[5] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *arXiv preprint arXiv:1905.00641*, 2019.

[6] C. N. Duong, K. Luu, K. G. Quach, and N. Le. Shrinkteanet: Million-scale lightweight face recognition via shrinking teacher-student networks. *arXiv preprint arXiv:1905.10620*, 2019.

[7] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, pages 87–102. Springer, 2016.

[8] 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, 2015.

[9] 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, 2016.

[10] B. Heo, M. Lee, S. Yun, and J. Y. Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. *arXiv preprint arXiv:1811.03233*, 2018.

[11] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[12] 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.

[13] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[14] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. 2008.

[15] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

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

[17] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[18] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou. Agedb: the first manually collected, in-the-wild age database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 51–59, 2017.

[19] 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.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[21] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs. Frontal to profile face verification

in the wild. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[22] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[23] Y. Xing, S. Liang, L. Sui, X. Jia, J. Qiu, X. Liu, Y. Wang, Y. Wang, and Y. Shan. Dnnvm: End-to-end compiler leveraging heterogeneous optimizations on fpga-based cnn accelerators. *arXiv preprint arXiv:1902.07463*, 2019.

[24] J. Yim, D. Joo, J. Bae, and J. Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017.

[25] Q. Zhang, J. Li, M. Yao, L. Song, H. Zhou, Z. Li, W. Meng, X. Zhang, and G. Wang. Vargnet: Variable group convolutional neural network for efficient embedded computing. *arXiv preprint arXiv:1907.05653*, 2019.

[26] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.