Factorizing and Reconstituting Large-kernel MBConv for Lightweight Face Recognition

Yaqi Lyu Jing Jiang Kun Zhang Yilun Hua Miao Cheng
Dahua Technology, Hangzhou, China
{lv_yaqi, jiang_qing1, zhang_kun7, hua_yilun, cheng_miao}@dahuatech.com

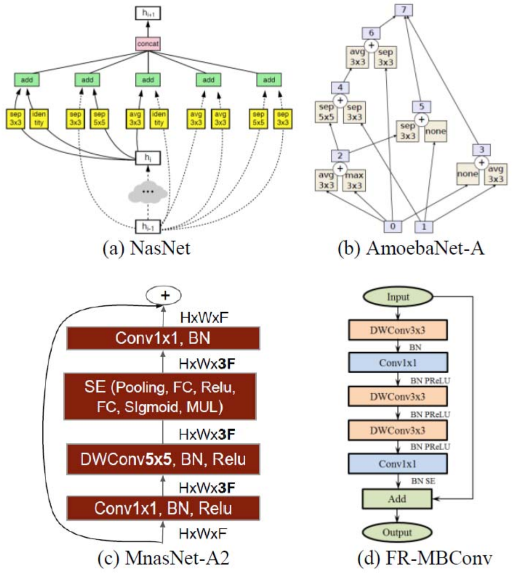
Abstract

In the past few years, Neural Architecture Search (NAS) has exhibited remarkable advances in terms of neural architecture design, especially on mobile devices. NAS normally use hand-craft MBConv as building block. However, they mainly searched for block-related hyperparameters, and the structure of MBConv itself was largely overlooked. This paper investigates that factorization and reconstitution can promote the efficiency of large-kernel MBConv and thus proposes FR-MBConv (Factorizing and Reconstituting large-kernel MBConv). Compared to large-kernel MBConv with the same receptive field, our FR-MBConv has fewer number of parameters and less computational cost, dramatically increased depth and nonlinearity. In addition, from the perspective of feature generation mechanism, FR-MBConv can be equivalent to more regular convolutions.

We combine FR-MBConv with MobileNetV3 [16] to build a lightweight face recognition model. Extensive experiments on face recognition benchmark demonstrate that our lightweight face recognition model outperforms the state-of-the-art lightweight model. Even on large scale face recognition benchmark IJB-B, IJB-C and MegaFace, our lightweight model also achieves comparable performance with large models.

1. Introduction

Convolutional neural networks (CNN) have made significant progress in the related applications of computer vision, such as image classification, object detection, visual object tracking, and semantic segmentation. As modern CNN models become increasingly deeper and larger, they also become slower, and require more computation [25] [33] [20] [9] [7]. Such increases in computational costs make it difficult to deploy state-of-the-art (SOTA) CNN models on resource-constrained platforms.



(a) NasNet (b) AmoebaNet-A

(c) MnasNet-A2 (d) FR-MBConv

Figure 1. Comparison of different blocks. (a) and (b) are search-based blocks. (c) and (d) are hand-craft blocks. NasNet, AmoebaNet and MnasNet illustrations are excerpted from respective papers. FR-MBConv is proposed in this paper.

Recently Neural Architecture Search (NAS) achieved the top performance on lightweight CNN design. In order to balance search space and search cost, NAS-based methods always draw prior from hand-craft CNNs. NAS can be summarized into two mainstream directions: cell-based search methods such as NasNet [4], PNAS [6], ENAS [14], and structural hyperparameter-based search methods such as MnasNet [27], ProxylessNAS [13], FBNet [3], and Single-path NAS [34]. The former only searches for the structure of cells (or blocks) and repeat them to build networks. The blocks searched are always multi-path and fragmented structure, and show poor performance-latency trade-off. The later directly uses hand-craft MBConv (Mobile Inverted Bottleneck Convolution, which was proposed in paper [26]), as building blocks and searches for structural hyperparameters in limited search space. Rather than optimizing the structure of block as cell-based methods, these methods simply search hyperparameters of block.

Hardware-aware NAS methods integrate latency into object function, and as a result these methods achieved the best performance-latency trade-off. Some blocks are shown in Figure 1.

Although the MBConv was extensively used in NAS-based methods, we discover that this block has following drawbacks:

(1) The computations of the two pointwise layers (i.e. expansion layer and projection layer) are much higher than the middle depthwise layer, but only the later makes contribution to receptive field. The property of low cost makes depthwise layer extraordinarily cheap to employ large kernels in MBConv.

(2) From the perspective of feature generation mechanism, the depthwise layer and the projection layer in an MBConv can be regarded as a regular convolution [1]. However, the expansion layer always follows the projection layer of the last MBConv, thus having little effect on feature generation except changing feature dimensions. This heavy expansion layer is a little wasteful.

(3) Modern hardware-aware efficient NAS methods searched for hyperparameters of MBConv, like kernel size, expansion factor, squeeze-and-excitation (SE) ratio, but the structure of layers was largely overlooked.

(4) Successful large CNNs tend to use small-kernel convolutions to design networks [25] [33] [20] [9] [7], but NAS networks searched [27] [13] [3] [34] include many large-kernel MBConv. Moreover, factorizing large kernels into small kernels cannot improve efficiency but hurt accuracy.

(5) As suggested in paper [37], large kernels do not always improve accuracy. For face recognition tasks, large kernel notably hurt accuracy, as shown in Figure 6. Although large kernels need more parameters and computations, the performance on CFP dataset [31] drops down quickly from 3x3 kernels to 7x7 kernels.

Considering above drawbacks, in this paper, we study the effect of factorization and reconstitution on large-kernel MBConv. We discover that factorizing inside block and feature generation mechanism guided reconstitution achieves more efficient performance than naïve large-kernel MBConv. Then we present a simple and efficient block called FR-MBConv (***F**actorizing and **R**econstituting **M**obile inverted **B**ottleneck **Conv**olution*), as an alternative to MBConv. Compared to MBConv with the same receptive field, FR-MBConv has fewer parameters and computations, considerably increased depth and nonlinearity. It is also easier to be trained from scratch and deployed without additional optimization on large kernels. From the perspective of feature generation mechanism , FR-MBConv can be equivalent to more regular convolutions, indicating stronger ability of feature representation. In addition, our FR-MBConv is a generic block, which can be used to replace MBConv in MobileNetV2-like network to get better performance- latency trade-off.

We combine FR-MBConv and MobileNetV3 [16], a recently proposed hardware-ware efficient network, to build a lightweight face recognition model. In order to make it easier to be deployed on mobile devices, we decrease the model size in terms of the feature embedding module and the SE integration strategy. Extensive experiments on face recognition benchmarks demonstrate that our lightweight face recognition model outperforms SOTA lightweight models, and achieves the best performance-efficiency trade-off. Our lightweight model achieves competitive performance compared with ResNet100-based large models on large scale face recognition benchmark IJB-B (94.5%), IJB-C (96.0%) and MegaFace (97.8%). Our contributions can be summarized as follows:

- We investigate the effect of factorization and reconstitution on large-kernel MBConv, and propose FR-MBConv, as a simple and efficient alternative to MBConv.
- We combine FR-MBConv and MobileNetV3 to build a lightweight face recognition model. We further explored feature embedding module and SE integration strategy to regulate model size.
- Experiments on several face recognition benchmarks indicate that our model is comparable with large models.

2. Related work

2.1. Hand-craft mobile building blocks

MobileNetV1 [1] introduced depthwise separable convolutions as an efficient replacement for traditional convolution layers. From the respect of feature generation mechanism, depthwise separable convolutions effectively factorize traditional convolution by separating spatial filtering. Depthwise separable convolutions consist of two separate layers: a 3x3 depthwise convolution for spatial filtering inside each channel and a 1x1 pointwise convolution for exchanging information across channels. Computations and parameters of MobileNetV1 are mostly expend on pointwise convolutions, in theory which is 31x and 70x times than depthwise convolutions respectively. In MobileNetV1 block, the time cost of the 1x1 convolution is 4.3x times than the 3x3 depthwise convolutions when testing on an Apple iPhone X [5]. MobileNetV2 [26] introduced the linear bottleneck and inverted residual structure, and proposed more efficient MBConv block. The structure of MBConv is a 1x1 expansion convolution followed by depthwise convolutions and a 1x1 projection layer. The input and output are connected with a residual connection if and only if they have the same number of channels. This structure maintains a compact representation at the input and the output, while expanding to a higher-dimensional feature space internally, which enlarges the proportion of computations of the depthwise convolution and increases the expressiveness of spatial filtering. MBConv is one of the best handcraft blocks and extensively utilized in NAS-based methods. MnasNet [27] and MobileNetV3 [16] drew large kernels and SE [20] based lightweight attention module into MBConv. The SE module is placed before the projection layer in the expansion feature space to maximize channelwise modulation. Most recently, MDConv (mixed depthwise convolution) [37] is proposed to mix up multiple kernel sizes in a single depthwise convolution to capture different types of patterns. As an alternative, in this paper we introduce convolutional factorization to MBConv to make it more efficient.

2.2. Hardware-aware efficient NAS

Based on reinforcement learning, MnasNet [27] used MobileNetV2 [26] as the baseline network structure and searched for structural hyperparameters. The search space of MnasNet include block related options like kernel size, expansion factor, SE ratio, and stage related options like numbers of channels and blocks. However, the structure of MBConv was disregard. Network searched use both 3x3 and 5x5 convolutions to have better accuracy-latency tradeoff. MobileNetv3 [16] used the NetAdapt [36] algorithm to search per layer for the number of filters. Furthermore, swish [28] was introduced to replace ReLU. Swish significantly improves the accuracy but increases latency cost, so it was used only in the deep layers. Differentiable NAS [13] [3] [34] methods train a single over-parameterized super-model network to prune a compact optimized architecture. These methods significantly reduce search cost, but also decrease search space to only block-related hyperparameters. ProxylessNAS [13] and FBNet [3] formulated the architecture searching problem to a multi-path selection problem. Besides, ProxylessNAS introduced 7x7 kernels to MBConv and FBNet introduced group convolutions to the pointwise layers. Search results show that large kernels frequently appear in the deep layers and group option is merely used. Single-Path NAS [34] encoded all architectural decisions based on shared convolutional kernel parameters, and drastically decreased the search cost. In summary, hardware-aware efficient NAS methods directly use hand-craft MBConv as building block, and mainly search for MBConv-related hyperparameters, especially differentiable NAS methods. However, all these methods do nothing with the structure of MBConv.

2.3. Lightweight Face Recognition

The face recognition models deployed locally on mobile devices are expected to be not only accurate but also small and fast. Lightweight network specifically designed for face recognition have been rarely researched. The most straight-forward way is combining above efficient networks with SOTA face recognition loss [22] [11] [39] [15]. ArcFace [22] utilized MobileNetV1 as backbone and the BN-Dropout-FC-BN structure to get the 512-D embedding feature. For the drawback, this structure increases the model size to 112M. MobileFaceNet [32] proposed global depthwise convolution (GDC) to make a lightweight feature embedding module. They employed the Conv1x1-BN-GDC7x7-FC-BN structure to get the 128-D embedding feature which greatly reduces model size (only 4M). However the computation of this structure is comparable with ArcFace, and relatively poor performance. We propose a feature embedding module in this paper, which achieves better performance-parameter trade-off than both above. ShrinkTeaNet [10] introduced a teacher-student learning algorithm to train lightweight face recognition model. With FR-MBConv, our lightweight model can be trained from scratch without any assistance.

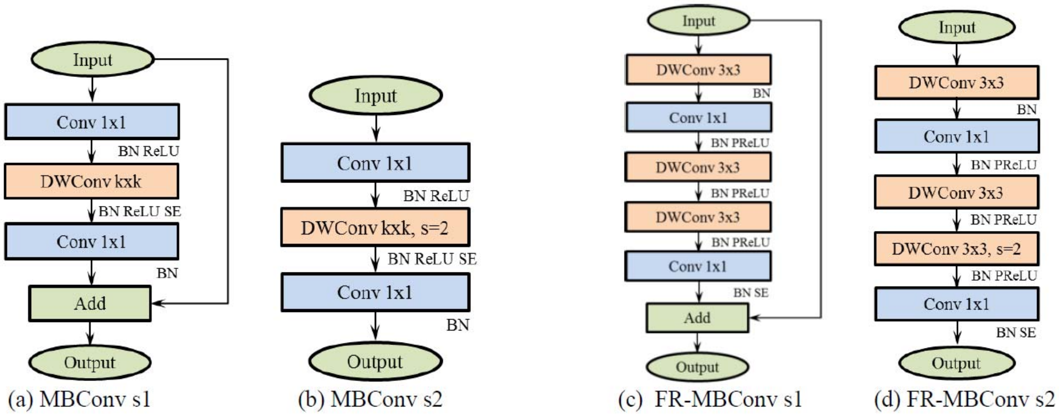
3. Proposed method

We propose FR-MBConv, as a simple and efficient alternative to MBConv. The FR-MBConv can be regard as a factorization-reconstitution version of large-kernel MBConv but more lightweight and powerful. Then we combine FR-MBConv with MobileNetV3 to build a lightweight face recognition model.

3.1. FR-MBConv: an efficient and powerful mobile block

The structures of our FR-MBConv are shown in Figure 2 (c) and (d). Similar to MBConv, the first 1x1 convolution is expansion layer, and the second 1x1 convolution is projection layer. The input and output are connected with a residual connection if and only if they have the same number of channels. The expansion convolution will be skipped if expand factor is 1. Each convolution layer is followed by a Batch Normalization [19] layer, and linear activation is applied after the first depthwise convolution and the last pointwise convolution. We set the stride of last depthwise convolution to 2 when needed. The difference is that, FR-MBConv employ three 3x3 depthwise convolutions without hyperparameter of kernel size.

Compared with MBConv in Figure 2 (a) and (b), FR-MBConv has the following advantages: (1) Less parameters and computation with the same receptive field; (2) More depth and nonlinearity inside block; (3) From the respect of feature generation mechanism, FR-MBConv can be regarded as more regular convolutions; (4) FR-MBConv is much easier to train from scratch, and deployment friendly, without additional optimization on 5x5 and 7x7 depthwise convolutions. In addition, our FR-MBConv is a



(a) MBConv s1 (b) MBConv s2 (c) FR-MBConv s1 (d) FR-MBConv s2

Figure 2. Blocks of MBConv and FR-MBConv. (a) and (b): The structures of MBConv block with stride 1 and 2, in which k is kernel size, and default nonlinearity is ReLU or swish. (c) and (d): Thee structures of our FR-MBConv block with stride 1 and 2, in which default nonlinearity is PReLU. SE refers to SE model with ratio 0.25.

generic block, which can be used to replace MBConv in MobileNetV2-like networks, or directly inserted in search space of NAS.

The main idea of FR-MBConv is factorizing and reconstituting large-kernel MBConv. In this section, we will discuss the advantages of factorization and reconstitution.

Factorization makes mobile block more efficient. For regular convolution, a large-kernel convolution can be factorized into several small-kernel convolutions with the same receptive field (RF), but the later has less computation and parameters, and could insert more nonlinearity. However, permitted to use different kernel size, networks searched tend to use large kernels in the deep layers. Inspired by regular convolution, we bring out a question: *Why didn't NAS factorize large kernels in the search process?*

For an MBConv, both the expansion layer and the projection layer are pointwise convolutions, which cannot increase receptive field. The receptive field of the middle depthwise convolution is equal to the whole MBConv. Formally, given the input tensor with shape (H, W, C), e and k represent expansion factor and kernel size of the depthwise layer of an MBConv. The FLOPs of pointwise layers denote as PWF:

$$PWF = 2e \cdot H \cdot W \cdot C^2 \tag{1}$$

The FLOPs of depthwise layer denote as DWF:

$$DWF = k^2 \cdot e \cdot H \cdot W \cdot C \tag{2}$$

The total FLOPs of MBConv are the sum of PWF and DWF. Large kernel size delivers larger receptive field, but more FLOPs. The ratio of PWF and DWF is $\frac{2C}{k^2}$, which means pointwise layers have heavier burden in computation. It's worth noting that kernel size only affects the computation of depthwise layer.

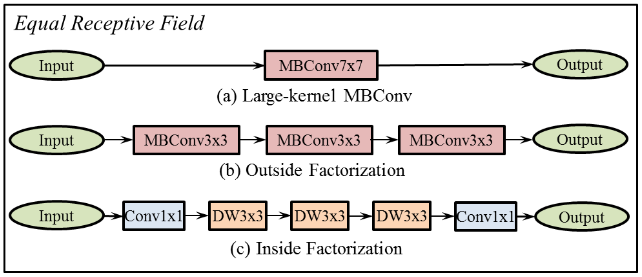


Figure 3. Comparison of different factorization strategy. (a): Original large-kernel MBConv. (b): Factorizing it outside block. (c): Factorizing it inside block. DW3x3 refers to 3x3 depthwise convolution.

In search space of modern NAS, MBConv is regarded as a complete component. As shown in Figure 3, a 7x7 MBConv can be factorized into three 3x3 small-kernel MBConv as shown in (b), and we call it outside factorization. On the contrary, inside factorization means only factorize the large-kernel depthwise convolution inside MBConv, as shown in (c). Table 1 shows the computation of each structure in Figure 3. Comparing (a) and (b), the total FLOPs of outside factorization is much higher than original MBConv, but DWF of the former is 0.55 times of the later, which means factorization works in depthwise layer. However, the PWF of outside factorization are 3 times of original MBConv, that's the reason why NAS don't factorize large kernels in search process.

Comparing (a) and (c), the PWF of inside factorization is equal to original MBConv, but its DWF is significantly reduced by 45%. It's worth noting that the number of parameters in depthwise layer is reduced as equal. In addition, inside factorization makes block deeper so that we can attach more nonlinearity units. This reduction means

Block	MBConv	Outside-fact	Inside-face
RF	7	7	7
PWF	$2e \cdot H \cdot W \cdot C^2$	$6e \cdot H \cdot W \cdot C^2$	$2e \cdot H \cdot W \cdot C^2$
PWF	$49e \cdot H \cdot W \cdot C$	$27e \cdot H \cdot W \cdot C$	$27e \cdot H \cdot W \cdot C$

Table 1. Comparison of structures in Figure 3. MBConv corresponds to structure (a), Outside-Fact and Inside-Fact mean outside factorization and inside factorization, corresponding to structure (b) and structure (c) respectively.

Block	MBConv	F-MBConv	FR-MBConv
RF	7	7	7
PWF	$2e \cdot H \cdot W \cdot C^2$	$2e \cdot H \cdot W \cdot C^2$	$2e \cdot H \cdot W \cdot C^2$
PWF	$49e \cdot H \cdot W \cdot C$	$27e \cdot H \cdot W \cdot C$	$(18e + 9) \cdot H \cdot W \cdot C$

Table 2. Comparison of structures in Figure 4. MBConv corresponds to structure (a), Outside-Fact and Inside-Fact mean outside factorization and inside factorization, corresponding to structure (b) and structure (c) respectively.

inside factorization works just like regular convolution. In summary, we can utilize inside factorization to make mobile block more efficient.

Reconstitution makes mobile block stronger. As mentioned above, after factorization of a 7x7 large-kernel MBConv, we obtain three 3x3 depthwise convolutions, thus we can insert three nonlinearities into block (refers to F-MBConv). However, for whole network consist of F-MBConv, the two pointwise layers are adjacent and three depthwise layers are adjacent, causing monotonous and redundant operation. In this section we introduce reconstitution to make block crisscross and make the utmost of both kinds of layers.

As shown in Figure 4, we first factorize the 7x7 depthwise convolution into three 3x3 depthwise convolutions inside the MBConv, as shown in (b) and (c). Table 2 lists the computation of each block in Figure 4.

From the view of feature generation mechanism, a regular convolution layer can be factorized into a depthwise layer following a pointwise layer. The middle depthwise layer and following pointwise layer in structure (b) can be factorized inversely to a regular convolution as structure (a). However after factorization, there is only one depthwise layer following a pointwise layer as before. To make block crisscross, the first depthwise convolution in structure (c) is moved to the font of the first pointwise layer, as shown in (d), which is the core structure of our FR-MBConv. We call this process as reconstitution. This reconstitution has two advantages. Firstly, after reconstitution the first and third depthwise layers are both followed by a pointwise layer, therefore we can factorize them inversely to regular convolutions, as shown in structure (e). Our FR-MBConv makes fully use of heavy pointwise convolutions for exchanging information across channels, indicating stronger ability of feature representation. Secondly, the first depthwise convolution was moved from expansion feature space to compact feature space, reducing its computation. As shown in Table 2, factorization and reconstitution both reduce DWF, while avoiding changing PWF. If expansion factor is 3 or 6, though factorization and reconstitution, DWF were reduced by 43% and 40% respectively.

To sum up, inside factorization and feature generation mechanism guided reconstitution are two useful tricks to promote the efficiency of large-kernel MBConv.

3.2. Lightweight Face Recognition

Our FR-MBConv is generic so that it is possible to replace MBConv with this block in any network. In this section, based on a SOTA hardware-aware efficient MobileNetV3-large, we built a lightweight face recognition model. To meet the requirements of the Track 1 of Lightweight Face Recognition Challenge [21], we made some modifications to the model. First, we fixed the input face image to 112x112, and substitute the first three layers of MobileNetV3-large [16] by a 3x3 regular convolution with 24 channels and stride 2. Then, all MBConv blocks are replaced by FR-MBConv with the same channels. We adjusted expansion factor of each block, and used PReLU as nonlinear activation. After that, the lightweight model was scaled up to match the FLOPs and model size request of Track 1. Different from EfficientNet [38], we kept the resolution of the input image, and only scaled up width and depth of model ($\alpha = 1.2, \beta = 1.1, \gamma = 1$). Our lightweight face recognition network is shown in Table 3.

In order to better deploy our lightweight network on resource-constrained mobile devices, we optimized the model size in two sides: **(a) Feature embedding module.** We proposed a DW3x3-BN-FC-BN structure to get the 512-D embedding feature, where DW3x3 is a 3x3 depthwise convolution with stride 2 and no padding. Our new feature embedding module achieves better computation-parameter trade-off than the embedding module of MobileFaceNet [32] and ArcFace [22]. The rest layers after last FR-MBConv were replaced by our feature embedding module. **(b) Squeeze-and-excited module integration strategy.** MobileNetV3 put SE module in expansion feature space, which leads to 8MB larger than MobileNetV2 in model size. We used PReLU and sigmoid as nonlinearity instead of ReLU and hard-sigmoid in MobileNetV3 block, and we find this setting make model easier convergence. Then we put SE module after projection layer with SE ratio fixed to 0.25, within compact feature space. The number of parameters in SE module could be decreased by 97% when expansion factor was 6. We employed SE module in every FR-MBConv owing its high efficiency.

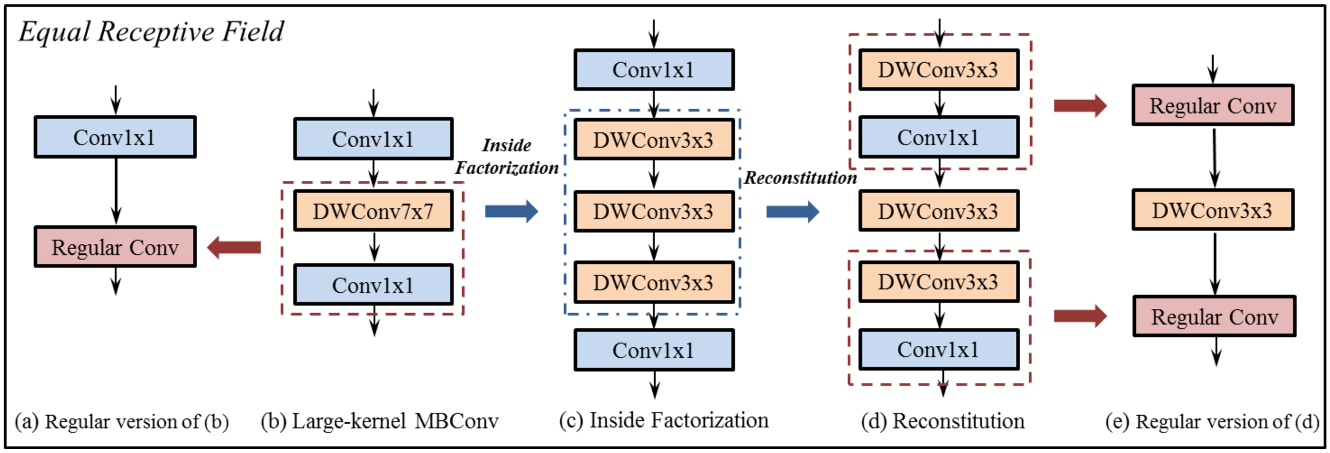


Figure 4. Comparison of different structures of blocks with the same receptive field. (b): A 7x7 MBConv. (c) and (d): The results after factorization and further reconstitution. (a) and (e): Equal structures of (b) and (d) in the respect of feature generation mechanism. We ignored all BN, SE, nonlinearity and shotcut connections.

Input	Block	Output	e	n	s
$112^2 \times 3$	Conv3x3	$56^2 \times 32$	-	1	2
$56^2 \times 32$	FR-MBConv	$56^2 \times 32$	1	2	1
$56^2 \times 32$	FR-MBConv	$28^2 \times 48$	4	5	2
$28^2 \times 48$	FR-MBConv	$14^2 \times 104$	6	6	2
$14^2 \times 104$	FR-MBConv	$14^2 \times 144$	6	3	1
$14^2 \times 144$	FR-MBConv	$7^2 \times 200$	6	5	2
$7^2 \times 200$	DW3x3	$3^2 \times 200$	-	1	2
$3^2 \times 200$	FC	512	-	1	

Table 3. Lightweight face recognition network proposed in this paper. The expansion factor of each block is e. Each line describes a stage, in which the block repeat n times. The first block of each stage has a stride s and all others use stride 1. DW3x3 is depthwise convolution with stride 2 and padding 0.

4. Experiments

4.1. Databases

Training set: MS-Celeb-1M [41] dataset contains 10M face images of 100k identities. A cleaned version of original MS-Celeb-1M with 5.1M face images of 93k identities is provided by Lightweight Face Recognition Challenge [21], where all face images are preprocessed to 112x112 by RetinaFace [23]. In this work we directly use this cleaned version, denote as MS1Mt1, as training set without any modification for a fair comparison.

Test set: Three small-scale face verification dataset LFW [12], CFP [31] and AgeDB [30], and two large-scale face recognition dataset IJB-B [8], IJB-C [2] and MegaFace [17] are utilized.

4.2. Implementation details

All experiments in this work are implemented based on Insightface [18] by MXNet [35]. We use ArcFace loss [22] with $s = 64$ and $m = 0.5$. All models in experiments use the weight initialization strategy described in [24] and were trained from scratch. We use SGD with a mini-batch size of 1024 on 8 GPUs (128 per GPU). The weight decay is 0.00005 and the momentum is 0.9. The learning rate starts from 0.1 and is divided by 10 at 260K, 340K and 360K iterations. The training process is finished at 380K iterations. All our convolutional layers use Batch- Normalization layers with average decay of 0.99. We only use horizontal flip augmentation at both training and testing stage.

4.3. Evaluation results

Results on LFW, CFP, AgeDB. In Table 4, we compare our model with other lightweight face recognition models. MobileNetV1 and MobileNetV2 are lightweight networks directly used for face recognition. MobileFaceNet-1G is a depth scaling version of MobileFaceNet with 1G FLOPs (y2 of insightface [18]). MobileNetV3-1G is a depth and channel scaling version of MobileNetV3 with 1G FLOPs, which is similar with our model except blocks are 3x3 and 5x5 MBConv, and nonlinear activations are ReLU and swish. We find that original MBConv is hard to train, and MobileNetV3 even cannot converge. On the contrary, our FR-MBConv-based networks converge quickly even though training from scratch.

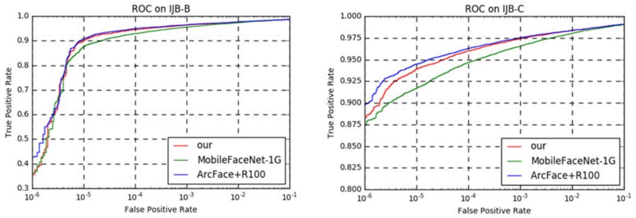
As shown in Table 4, on the LFW, CFP and AgeDB, our network achieves best performance. The verification rate of our network is boosted to 98.54%, 3.4% higher than distillation model ShrinkTeaNet-MFNR on CFP dataset, which means the error rate is reduced by 70%. With similar

Method	Size	Train-Set	LFW	CFP	AgeDB
Res50+SphereFace [39]	167	CASIA	99.11	94.38	91.70
Res50+CosFace [15]	167	CASIA	99.51	95.44	94.56
Res50+ArcFace [22]	167	CASIA	99.53	95.56	95.15
MobileNetV1 [1]	14.1	MSIMv2	99.53	93.81	96.30
MobileNetV2 [26]	8.6	MSIMv2	99.42	91.67	95.28
MobileFaceNet [32]	4.8	MSIMv2	99.45	92.11	96.17
ShrinkTeaNet [10]	14.9	MSIMv2	99.77	95.14	97.63
MobileFaceNet-1G	8.25	MSIMt1	99.73	97.73	97.33
MobileNetV3-1G	21.5	MSIMt1	-	-	-
Our	19.8	MSIMt1	**99.80**	**98.54**	**98.11**

Table 4. Verification performance (%) of SOTA models on LFW, CFP and AgeDB. Results of the first three are from ArcFace [22]. MS1Mv2 is another cleaned version of MS1M. The unit of size is MB.

Method	Size	Train-Set	IJB-B	IJB-C
Res50 [29]	-	VGG2	78.4	82.5
SENet50 [29]	-	VGG2	80.0	84.0
MN-vc [**?**]	-	VGG2	83.1	86.2
Res50+DCN [40]	-	VGG2	85.0	86.7
ArcFace+Res50 [22]	167	VGG2	89.8	92.1
ArcFace+Res100 [22]	250	MS1Mv2	**94.9**	**96.3**
MobileNetV1 [1]	14.1	MSIMv2	90.9	93.0
MobileNetV2 [26]	8.6	MSIMv2	88.3	90.7
MobileFaceNet [32]	4.8	MSIMv2	89.6	91.8
ShrinkTeaNet [10]	14.9	MSIMv2	92.3	94.0
MobileFaceNet-1G	8.25	MSIMt1	92.9	94.6
Our	19.8	MSIMt1	**94.5**	**96.0**

Table 5. Verification (%) TAR (@FAR=1e-4) on IJB-B and IJB-C datasets. Results are from respective papers. The unit of size is MB.



(a) IJB-B

(b) IJB-C

Figure 5. ROC curves of 1:1 verification protocol on the IJB-B and IJB-C dataset.

Method	Train-Set	Id(%)	Ver(%)
FaceNet [11]	-	70.49	86.47
CosFace [15]	-	82.72	96.65
R100+CosFace [22]	MS1Mv2	80.56	96.56
R100+ArcFace [22]	MS1Mv2	81.03	96.98
MobileFaceNet [32]	MS1Mv2	-	90.16
Our	MS1Mt1	**80.29**	**96.47**
R100+CosFace(R) [22]	MS1Mv2	97.91	97.91
R100+ArcFace(R) [22]	MS1Mv2	98.35	98.48
MobileFaceNet(R) [32]	MS1Mv2	-	92.59
ShrinkTeaNet(R) [10]	MS1Mv2	95.64	-
Our(R)	MSIMt1	**97.80**	**97.94**

Table 6. Face identification and verification on MegaFace. Results are from respective papers. "Id" refers to the rank-1 face identification accuracy with 1M distractors, and "Ver" refers to the face verification TAR at 10-6 FAR. "R" refers to the refined version of MegaFace (according to [22]). The unit of size is MB.

FLOPs, our network significantly reduces error rate by 36% and 29% than MobileFaceNet-1G respectively on CFP and AgeDB. Its worth noting that with similar FLOPs, model size of our network is 11.55MB large than MobileFaceNet-1G, among which most parameters spend on SE module.

Results on IJB-B and IJB-C. In Table 5 we compare our network with state-of-the-art models at TAR (@FAR=1e-4) on large-scale datasets IJB-B and IJB-C. Our network is 1.6% and 1.4% higher than MobileFaceNet-1, and achieves best performance among lightweight models. Our network improves the TAR(@FAR=1e-4) of lightweight model to 0.945 and 0.960 on IJB-B and IJBC respectively. Besides, our model is comparable with SOTA large model ArcFace with ResNet100, which is 13.6 times larger than our model. In Figure 5, we show the full ROC curves of our network with MobileFaceNet-1G and ArcFace on IJB-B and IJB-C.

Results on MegaFace. In Table 6 we compare our network on MegaFace dataset under the large protocol. Our network achieves the best performance among lightweight networks, 6.3% higher verification rate than MobileFaceNet on the original dataset, and 2.2% higher identification rate than newly distillation based ShrinkTeaNet-MFNR on the refined dataset. In addition, our model is also comparable with ResNet-100-based ArcFace model, showing great potential of lightweight models.

4.4. Ablation study

In Table 7, we first compare our FR-MBConv with MBConv. MBConv with different kernel size directly replace FR-MBConv in Table 3. It's worth noting that we use MBConv with the same kernel size and SE settings in each network for simplicity. FR-MBConv kxk denotes the structure of MBConv kxk after factorizing and reconstituting. Obviously FR-MBConv 7x7 is the base block in this paper. Comparing with original 7x7 MBConv, our 7x7 FR-MBConv reduces FLOPs and model size by 11% and 50% respectively. It's clear that our FR-MBConv has a better performance-efficiency trade-off than MBConv. As shown in Figure 6, FR-MBConv 3x3 is the same block as MBConv 3x3 except the location of SE module, which leads to half of model size smaller and 0.2% lower verification rate. Large-kernel MBConv requires more computations and parameters, but notably hurt performance on face recognition task. On the contrary with factorization and reconstitution, our FR-MBConv can benefit from additional depthwise convolutions.

Method	FLOPs	Size	LFW	CFP	AgeDB
MB3x3	1019	38.6	**99.82**	**98.74**	**98.15**
MB5x5	1087	39.4	99.80	98.53	98.07
MB7x7	1205	40.6	99.73	97.84	97.78
FR3x3	1009	19.0	99.80	98.46	98.10
FR7x7	1022	19.8	99.80	98.54	98.11

Table 7. Comparison of different blocks version of our network. "MB kxk" refers to MBConv with kernel kxk. "FR kxk" refers to FR-MBConv corresponding to MBConv kxk. FLOPs are computed following Lightweight Face Recognition Challenge. The unit of size and FLOPs are MB and M respectively.

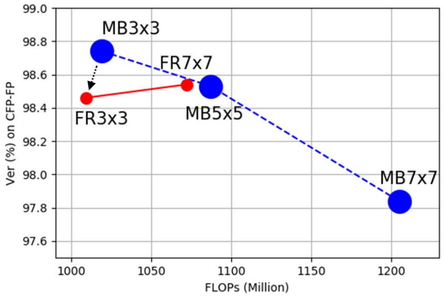


Figure 6. FLOPs vs. Ver. on CFP dataset. Each point represents our lightweight face recognition network with different block, where model size is represented by point size. Blue points denote MBConv with kernel size k, and each model size is 40MB. Red points denote FR-MBConv, and each model size is 20MB.

In Table 8, we compare different feature embedding module with our network. FC, GDC and S2FC indicate the feature embedding module of ArcFace, MobileFaceNet and our network respectively. It is worth noting that dropout layer of ArcFace is removed for fast convention. The FLOPs of network with our module is less than FC and GDC, and number of parameters is 44% of network with FC. FC gets best result on CFP with most parameters and FLOPs. Although with lowest model size, computation of GDC is comparable with FC. Furthermore, GDC is much worse than our S2FC on more challenge dataset CFP and AgeDB, indicating the drawback of compact feature dimension. It is obvious to see that our S2FC achieves better accuracy-model size trade-off than FC and GDC.

Finally, we compare SE integration strategies on face recognition task and results are reported in Table 9. SE-Pre denotes projection layer followed by SE module just like MobileNetV3, while SE-Post denotes SE module followed by projection layer, as our network. On LFW and AgeDB, performance of SE-Pre and SE-Post is comparable. On CFP dataset, SE-Pre achieve better recognition rate than SE-Post with 50% more parameters and slightly higher FLOPs. It's clear to see that our SE-Post integration strategy can reduce model size significantly with minor performance degradation.

Design	FLOPs	Size	LFW	CFP	AgeDB
FC-512	1030	35.4	99.77	**98.66**	97.95
GDC-128	1030	17.0	**99.82**	98.39	98.02
S2FC-512	1022	19.8	99.80	98.54	**98.11**

Table 8. Comparison of different feature embedding module. FC denotes BN-FC-FB structure of ArcFace. GDC denotes Conv1x1-BN-GDC7x7-FC-BN structure. S2FC denotes our DW3x3-FC-BN structure. FLOPs are computed following Lightweight Face Recognition Challenge. The unit of size and FLOPs are MB and M respectively.

Design	FLOPs	Size	LFW	CFP	AgeDB
SE-Pre	1032	39.4	99.78	**98.66**	98.10
SE-Post	1022	19.8	**99.80**	98.54	**98.11**

Table 9. Comparison of different feature embedding module. FC denotes BN-FC-FB structure of ArcFace. GDC denotes Conv1x1-BN-GDC7x7-FC-BN structure. S2FC denotes our DW3x3-FC-BN structure. FLOPs are computed following Lightweight Face Recognition Challenge. The unit of size and FLOPs are MB and M respectively.

5. Conclusion

In this paper, we research the effect of factorization and reconstitution on large-kernel MBConv. We discover that factorizing inside block and feature generation mechanism guided reconstitution achieves more efficient than naïve large-kernel MBConv. Thus, we proposed a simple and efficient block, namely FR-MBConv, to replace MBConv. Compared to MBConv, at same receptive field, FR-MBConv has less parameters and computational cost, considerably increased depth and more nonlinearity. It is also easier to be trained and deployed. We combined generic FR-MBConv with MobileNetV3 to build a lightweight face recognition model. In order to decrease model size, we further explored feature embedding module and squeeze-and-excite integration strategy. Extensive experiments on both small-scale and large-scale datasets demonstrated that our lightweight face recognition network has achieved state-of-the-art performance.

References

[1] B. Chen D. Kalenichenko W.Wang T. Weyand M. Andreetto A. G. Howard, M. Zhu and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR, abs/1704.04861*, 2018.

[2] J. A. Duncan N. Kalka T. Miller C. Otto A. K. Jain W. T. Niggel J. Anderson B. Maze, J. Adams and J. Cheney. Iarpa janus benchmark–c: Face dataset and protocol. *ICB*, 2018.

[3] P. Zhang Y. Wang F. Sun Y. Wu Y. Tian P. Vajda Y. Jia B. Wu, X. Dai and K. Keutzer. Fbnet: Hardware-aware efficient

convnet design via differentiable neural architecture search. *CVPR*, 2019.

[4] J. Shlens B. Zoph, V. Vasudevan and Q. V. Le. Learning transferable architectures for scalable image recognition. *CVPR*, 2018.

[5] Vakunov A et al Bazarevsky V, Kartynnik Y. Blazeface: Sub-millisecond neural face detection on mobile gpus. *arXiv preprint arXiv:1907.05047*, 2019.

[6] J. Shlens W. Hua L. Li L. Fei-Fei A. L. Yuille J. Huang C. Liu, B. Zoph and K. Murphy. Progressive neural architecture search. *ECCV*, 2018.

[7] V Vanhoucke AA Alemi C Szegedy, S Ioffe. Inception-v4, inception-resnet and the impact of residual connections on learning. *AAAI*, 2017.

[8] A. Blanton B. Maze J. C. Adams T. Miller N. D. Kalka A. K. Jain J. A. Duncan C. Whitelam, E. Taborsky and K. Allen. Iarpa janus benchmark-b face dataset. *CVPRW*, 2017.

[9] Sergey Ioffe Jon Shlens Christian Szegedy, Vincent Vanhoucke and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CVPR*, 2016.

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

[11] D. Kalenichenko F. Schroff and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *CVPR*, 2015.

[12] T. Berg G. B. Huang, M. Ramesh and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *Technical report*, 2007.

[13] L. Zhu H. Cai and S. Han. Proxylessnas: Direct neural architecture search on target task and hardware. *ICLR*, 2019.

[14] B. Zoph Q. V. Le H. Pham, M. Y. Guan and J. Dean. Efficient neural architecture search via parameter sharing. *ICML*, 2018.

[15] Z. Zhou X. Ji Z. Li D. Gong J. Zhou H. Wang, Y. Wang and W. Liu. Cosface: Large margin cosine loss for deep face recognition. *CVPR*, 2018.

[16] Chu G et al Howard A, Sandler M. Searching for mobilenetv3. *arXiv preprint arXiv:1905.02244*, 2019.

[17] D. Miller I. Kemelmacher-Shlizerman, S. M. Seitz and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. *CVPR*, 2016.

[18] InsightFace. https://github.com/deepinsight/insightface.

[19] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML*, 2015.

[20] L. Shen J. Hu and G. Sun. Squeeze-and-excitation networks. *CVPR*, 2018.

[21] Debing Zhang Yafeng Deng Xiang Lu Song Shi Jiankang Deng, Jia Guo and Stefanos Zafairiou. Lightweight face recogonition challenge. *ICCV*, 2019.

[22] Xue Niannan Jiankang Deng, Jia Guo and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *CVPR*, 2019.

[23] Yuxiang Zhou Jinke Yu Irene Kotsia Jiankang Deng, Jia Guo and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *Arxiv*, 2019.

[24] S. Ren K. He, X. Zhang and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *ICCV*, 2015.

[25] S. Ren K. He, X. Zhang and J. Sun. Deep residual learning for image recognition. *CVPR*, 2016.

[26] M. Zhu A. Zhmoginov M. Sandler, A. G. Howard and L. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. mobile networks for classification, detection and segmentation. *CVPR*, 2018.

[27] R. Pang V. Vasudevan M. Tan, B. Chen and Q. V. Le. Mnasnet: Platform-aware neural architecture search for mobile. *CVPR*, 2019.

[28] B. Zoph P. Ramachandran and Q. V. Le. Searching for activation functions. *Searching for activation functions*, 2017.

[29] W. Xie O. M. Parkhi Q. Cao, L. Shen and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. *FG*, 2018.

[30] C. Sagonas J. Deng I. Kotsia S. Moschoglou, A. Papaioannou and S. Zafeiriou. Agedb: The first manually collected in-the-wild age database. *CVPRW*, 2017.

[31] C. Castillo V. M. Patel R. Chellappa S. Sengupta, J.-C. Chen and D. W. Jacobs. Frontal to profile face verification in the wild. *WACV*, 2016.

[32] Xiang Gao Sheng Chen, Yang Liu and Zhen Han. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. *CCBR*, 2018.

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[34] Wang D et al Stamoulis D, Ding R. Single-path nas: Designing hardware-efficient convnets in less than 4 hours. *arXiv preprint arXiv:1904.02877*, 2019.

[35] Y. Li M. Lin N. Wang M. Wang T. Xiao B. Xu C. Zhang T. Chen, M. Li and Z. Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*, 2015.

[36] B. Chen X. Zhang A. Go M. Sandler V. Sze T. Yang, A. G. Howard and H. Adam. Netadapt: Platform-aware neural network adaptation for mobile applications. *ECCV*, 2018.

[37] Mingxing Tan and Quoc V. Le. Mixnet: Mixed depthwise convolutional kernels. *BMVC*, 2019.

[38] Le Q V Tan M. Efficientnet: Rethinking model scaling for convolutional neural networks. *ICML*, 2019.

[39] Z. Yu M. Li B. Raj W. Liu, Y. Wen and L. Song. Sphereface: Deep hypersphere embedding for face recognition. *CVPR*, 2017.

[40] S. Li W. Xie and A. Zisserman. Comparator networks. *ECCV*, 2018.

[41] Y. Hu X. He Y. Guo, L. Zhang and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. *ECCV*, 2016.