Effective methods for lightweight image-based and video-based face recognition

Yidong Ma
Bitmain
Beijing, China
yidong.ma@bitmain.com

Abstract

Face recognition has achieved significant advances with the rise of deep convolutional neural networks (CNNs) and the development of large annotated datasets. However, how to design deep models in lightweight face recognition is still a challenge when aiming at mobile and embedded devices. In this paper, we focus on recent efficient CNN architectures, speedup skills and reduction methods to design models for lightweight face recognition. We combine octave convolution with MobileNet and ResNet for those models sensitive to computation complexity, replace feature output layer for those models sensitive to memory and explore network scaling for more powerful representation. Further, we extract a subset from the whole training dataset to speed up the performance evaluation of different models. We provide a scaling method on MobileFaceNet to boost the performance with the limit of computational cost, and propose a simple supplementary method for average pooling which throws up those noise frames based on the cluster information in video face recognition. With the upper bound of 1G FLOPs computation complexity and 20MB model size, our best model achieves 99.80% accuracy on LFW, 98.48% on AgeDB, 98% on CFP-FP and 97.67% TAR@FAR 10^{-6} on MegaFace.

1. Introduction

Deep learning has been widely used in many applications with impressive performance in recent years. Face recognition also has achieved significant advances on various datasets such as LFW[18] and MegaFace[21]. However, for applications in mobile and embedded devices such as face unlock, face login and mobile payment, it is difficult to deploy very deep nerual networks or use test augmentation methods of the competition to boost accuracy because of consumption of memories and computational resources. For common visual recognition, several efficient CNN architectures have been proposed, for example, MobileNet[16][29], SqueezeNet[19], ShuffleNet[39][26]. Also, there are some lightweight face recognition models proposed, such as ShiftFaceNet[37] and MobileFaceNet[3]. But reducing the model size always degrades the performance, and it is hard to make a tradeoff decision without a comprehensive understanding about the effectiveness of CNNs.

In this paper, we utilize several recent efficient neural network architectures, speedup skills and reduction methods to design different lightweight CNN models for image-based face recognition and video-based face recognition. Octave convolution[4], bottleneck and channel scaling are adopted to reduce FLOPs, average pooling for the last feature map and smaller embedding size are adopted to reduce the number of parameters, and the existing lightweight CNN models are used as backbone models for both. We also investigate how to scale up MobileFaceNet that can achieve better accuracy and efficiency. Then a small training dataset is constructed to evaluate these models and select better models. In video face recognition, average pooling is a commonly adopted strategy to aggregate features considering all face frames. So we propose a simple supplementary method for average pooling aiming at removing some noisy faces.

The major contributions of this work can be summarized as follows:

- We implement different lightweight CNNs with recent efficient architectures and effective model reduction methods.
- We provide a scaling method on MobileFaceNet for boosting performance and a simple supplementary method for feature aggregation.
- Experiments on LFW[18], AgeDB[27], CFP-FP[31], MegaFace[21] and deepglint-light[8] show that the selected model has comparable performance, and the supplementary method has a slight improvement on iQIYI-VID-light[8].

2. Related Work

Since AlexNet[22] won the ImageNet competition, CNNs has shown its power in common visual recognition tasks. Going deeper and bigger makes CNNs increasingly more accurate, but with large parameters and high memory consumption. Recently, lots of works focus on designing efficient architectures and how to strike an optimal balance between accuracy and speed. MobileNetV1[16] uses depthwise separable convolution, MobileNetV2[29] is based on an inverted residual structure with linear bottleneck, SqueezeNext[12] optimizes SqueezeNet[19] by simulating its performance on a multi-processor embedded system, ShuffleNet[39] utilizes pointwise group convolution and channel shuffle to reduce computation cost and ShuffleNetV2[26] gives practical guidelines for efficient network design. CondenseNet[17] combines dense connectivity with learned group convolution. [23][32] utilizes global average pooling over feature maps which reduces large numbers of parameter and is less prone to overfitting. [15] firstly proposes a bottleneck design, OctConv[4] can boost accuracy while reducing memory and computational cost. Also, neural architecture search (NAS) has made remarkable progress in producing models that transcend handcraft models. EfficientNet[34] studies CNNs scaling for width, depth and resolutions to meet different resource constraints. MNASNet[33] incorporates model latency to balance between accuracy and latency on mobile phones.

For image-based face recognition, many works [30][24][25][36][35] [7] focus on loss function to further improve the discriminative power of the model. To address issues of deployability, there are also many approaches proposed recently. ShiftFaceNet[37] presents a parameter-free, FLOP-free "shift" operation to reduce parameters. MobileFaceNet[3] replaces the global average pooling layer with a global depthwise convolution layer (GDConv). Compressing pretrained network methods like [20][11][10] use knowledge transfer and distillation to train a student model studying from a teacher model. [40] intergrates NAS technology info face recognition to customize a more suitable network which requires a lot of training time and computing capability. In this paper, we build models by prior knowledge with current practicable methods and construct a small training dataset for quickly performance evaluation.

For video-based face recognition, the key issue is to build an appropriate representation of the faces in video, effectively intergrating the information across different frames together and discarding noisy information. The most commonly adopted strategies may be average pooling and maxpooling[2][5][28]. To seek for an adaptive aggregation approach, Neural Aggregation Nework[38] is designed to adaptively calculate the weights for each of frames. Recently, C-FAN[13] automatically learns to retain salient face features with high quality scores while suppressing features with low quality scores. In this paper, we focus on a cheap computation method without neural networks.

3. Approach

In this section, we will describe how we build efficient models under the limitation of memories and computational resources. Following to LSR2019[8], we set the upper bound of computational complexity to 1G FLOPs, the upper bound of model size to 20MB and the upper bound of the feature dimension to 512. Quantization methods are not discussed in this paper. Meanwhile, we introduce our scaling method for MobileFaceNet and supplementary method of average pooling.

3.1. Efficient Models

FLOPs reduction. Convolution layers are computationally more efficient than fully connected layers (FC) because neurons receive only a restricted subarea of the previous layer with the same shared weights, however, when CNNs goes deeper and wider, it still need large computational resources and memories, so the variants of CNNs aiming at efficient computation and limited storage are proposed. Depthwise convolution and group convolution are frequently referred to, as they do not need entire channels of the previous layer when a kernel matrix applys to. So depthwise convolution and group convolution are useful for designing CNNs. Octave convolution operation is proposed to replace the regular convolution operation as it can reduce spatial redundancy. The convolutional feature maps are factorized into two groups at different spatial frequencies hence the resolution for low frequency maps can be reduced, saving both storage and computation. In this paper, we choose ResNet and MobileFaceNet as base models to modify. We put group convolution into bottleneck in ResNet and MobileFaceNet is already equipped with depthwise convolution. Following [4], we use average pooling and up-sampling operation via nearest interpolation to implement octave convolution. It can directly replace the regular convolution in ResNet without special adjustment and for depthwise convolution case, the information exchange paths are eliminated, leaving only two depthwise convolution operations. The built models are called Oct-ResNet-Neck, Oct-ResNet and Oct-MobileFaceNet.

Model Size reduction. For common CNNs, the early layers cost large computational resources and the last layers like FC need large numbers of parameter. Several previous works[32][23] already use global average pooling and 1×1 convolution to replace FC, and the recent global depthwise convolution[3](GDC) further improves global average pooling by learning different importances at different spatial positiions. So we will use GDC to replace FC if the

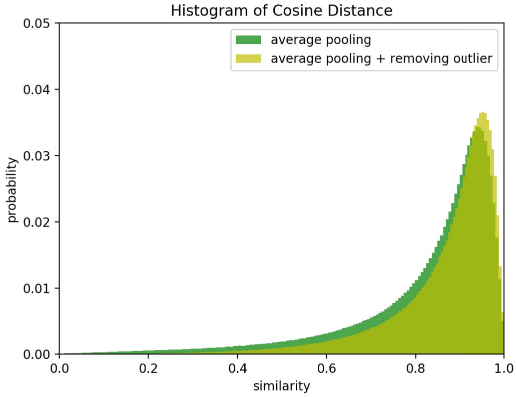

Figure 1. Green: after the average pooling, yellow: after removing outliers .

candidate model is oversize.

Both reduction. Reducing embedding size, scaling network depth and width can reduce the computation complexity and the number of parameter simultaneously. MobileFaceNet's last convolutional output has a stride of 16, following a GDC to get the final feature embedding. We keep the first convolution layer unchanged which has a stride of 2, and scale the block number of rest stages which have different resolutions i.e. 56, 28, 14 and 7. Our scaling method will focus on the high-level features which contains stronger semantic information, becasuse we find that increasing the computation complexity of previous stages gains a slower improvement. So we modify Y2[6], a variant of mobilefacenet, with different stage's block number of different resolutions to get different candidate models.

3.2. Feature Aggregation

Feature aggregation method should choose more discriminative faces and prevent poor faces from jeopardizing the recognition. The commonly adopted average pooling strategy is equal to find a centroid of all face feature vectors, i.e. KMeans. Actually, there is no difference for KMeans to choose between euclidean distance and cosine simlariy distance if the feature vectors are normalized. So, the average pooling can be an effective method in many works. Our supplementary method is simple: the poor faces which have adverse impacts on face recognition will deviate from the cluster center, so it is not necessary to use all faces to do average pooling. As shown in Figure 1, the green area represents the histogram of the distances to the cluster centers of faces after the average pooling. There are still many faces which have low similarity with their cluster centers and we call them outliers. Based on these similarity information, we can simply throw up those outliers which contribute little to or affect the aggregated feature. The yellow area shows that after removing outliers the rest faces are more compact with each other and the new aggregated feature is closer to the rest faces. This method almost cost no computing resources.

Model	Stage1	Stage2	Stage3	Stage4
Yoa	1.24	1.1	1.35	1.72
Yob	1.24	0.57	1.47	4.64
Yoc	1.48	1.63	0.88	2.09
Yod	1.0	0.57	1.47	4.64
Yoe	1.0	0.57	0.94	8.46

Table 1. All models are scaled from Y2 using different ratio-to-origin for each stage.

4. Experiments

4.1. Implementation Details

Our training dataset is cleaned MS-Celeb-1M[14] preprocessed by [6] which has 5.1M images of 93K identities with dozens to hundreds of faces in each identity. All face images are preprocessed to the size of 112×112 by the five facial landmarks predicted by RetinaFace[9]. LFW[18], AgeDB[27] and CFP-FP[31] are our face verification datasets. To quickly select the best model, we extract a subset of MS-Celeb-1M called Retina-S and expect the relative performance of different trained models on Retina-S can match that on cleaned MS-Celeb-1M. We filter the dataset with the upper bound of face number, and an identity will be discarded from training set if its number of face images is over a threshold set to 50. So Retina-S dataset has enough identities (1.1M) and fewer face images (42K). This step prevents from quick convergence and bad generalization.

With the upper bound of computational complexity and model size, we find original Y2[6] is already small, so we scale up Y2 by adjusting the block number of stage of different resolutions as Section 3.1. For ablation studies, we build different models as shown in Table 1 which have different scaling ratios for each stage. All variants of MobileFaceNet have a 256-D embedding feature. Based on ResNet[15], the block numbers of Oct-ResNet-Neck model are (3, 8, 36, 3) and the bottleneck module contains a 1×1 convolution layer for reducing channels following a group convolution layer and a 1×1 convolution layer for expanding channels, we use GDC as the feature output layer, set embedding size to 256 and add octave convolution to further reduce the FLOPs so that we do not change the original channel numbers. Without the bottleneck module, we get a regular Oct-ResNet model. We set embedding size to 192 to get Oct-ResNet-A and scale up the depth to get Oct-ResNet-B. We combine depthwise convolution and octave convolution to get Oct-MobileFaceNet, and scale up the depth or width to get two

Model	Size	FLOPs	LFW	AgeDB	CFP-FP
Yod	15MB	992M	**99.71**	96.18	97.21
Yoa	11MB	985M	99.70	96.35	**97.24**
Yob	16MB	993M	99.70	96.03	97.11
Y2	7.5M	768M	99.68	95.85	96.75
Yoc	9.1MB	939M	99.66	96.00	96.77
Yoe	19MB	898M	99.66	96.16	97.02
Oct-ResNet-A	19MB	856M	99.65	95.63	96.31
Oct-ResNet-B	20MB	976M	99.65	95.90	96.70
Oct-MobileFaceNet-A	19MB	963M	99.65	**96.38**	96.91
Oct-MobileFaceNet-B	15MB	897M	99.60	95.75	96.05
Oct-ResNet-Neck	17MB	963M	99.51	95.75	96.92
MNAS-1.25	19MB	870M	99.40	94.70	95.72

Table 2. Performance (%) comparison of different models trained on Retina-S without test augmentation.

Model	Size	FLOPs	LFW	AgeDB	CFP-FP
ShiftFaceNet[37]	3.1MB	-	96.00	-	-
TD-student[11]	2.3MB	-	99.27	94.25	-
MobileFaceNet[7]	3.9MB	-	99.50	95.91	88.94
MobileFaceNet[3]	4.0MB	419M	99.55	96.07	-
LResNet34E-IR[7]	131MB	8.3G	99.65	97.70	92.12
ShrinkTeaNet-MFNR[10]	3.73M	-	99.77	95.14	97.63
NAS A[40]	-	-	99.80	-	-
Ours	11MB	985M	**99.80**	**98.00**	**98.48**

Table 3. Performance (%) comparison with previous methods on LFW, AgeDB, CFP-FP.

version. For MNASNet, we scale up original network with a factor of 1.25 and modify the last channel number.

We train all candidate models on eight 1080Ti GPUs and set the batch size to 110 per GPU. We adopt Additive Anguler Margin Loss[7] and use cosine annealing learning rate, and it starts from 0.1 then reduces to 1e-4 in the 26 epochs. We set momentum to 0.9 and weight decay to 5e-4. After comparing the performance of models trained on the Retina-S, we set the total epochs to 96 and use the cleaned MS-Celeb-1M to train the selected model. For image-based face recognition, we test on large-scale image dataset (e.g. MegaFace[21] and deepglint-light[8]), For video-based face recognition, we take the iQIYI-VID[1] as our test set. We remove a fixed proportion (set to 0.2) of faces based on the cosine distance between the face feature vectors and the cluster center.

Model	Size	Id(%)	Ver(%)
FaceNet[30]	-	70.49	86.47
CosFace[35]	-	82.72	96.65
TD-student[11]	2.3MB	-	-
MobileFaceNet[3]	4.0MB	-	90.16
MobileFaceNet,R[3]	4.0MB	-	92.59
LResNet34E-IR,R[7]	131MB	96.59	**98.92**
ShrinkTeaNet-MFNR[10]	3.73M	-	95.64
NAS A[40]	-	-	-
Ours	11MB	**97.67**	98.56

Table 4. Face identification and verification evaluation of different models on MegaFace Challenge1. ”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 data refinement on both probe set and 1M distractors.

4.2. Evaluation Results

Results on LFW, CFP-FP, AgeDB In Table 2, we observe the scaled models have better performance on LFW, CFP-FP and AgeDB. Yoa, Yob and Yod achieve above 99.7% on LFW, and they all scale up high-level (i.e. stage 3 and stage 4) in Table 1. Although those models with octave convolution are better than MNAS-1.25, this reduction method drop the performance. We think this happens because the low frequency component damages the feature representation. We use Yoa to train more epochs, and it can achieve 99.80% accuracy on LFW, 98.48% on AgeDB and 98% on CFP-FP which shows comparable accuracy with previous methods (see Table 3).

Results on MegeFace In Table 4, our model achieves the best identification and verification performance outperforming other lightweight models. And, our model surpassing the LResNet34E-IR on identification with only 8.4% parameters and 11.6% computational cost. Our model can be an efficient model in image-based face recognition.

Results on deepglint-light and iQIYI-VID-light In LFR19 challenge, we use Yod to train more epochs, and we reach 86.335% at 10^{-8} FPR on deeplint-light. On iQIYI-VID-light, we achieve 57.169% at 10^{-4} FPR using the proposed supplementary method which is 0.62% higher than that using average pooling only.

5. Conclusion

In this paper, we implement different lightweight face recognition models with recent efficient architectures and model reduction methods. Then we provide a scaling method in image-based face recognition and a simple supplementary method in video-base face recognition. Our experiments show that our lightweight model has comparable accuracy and our supplementary method can enhance the face feature with cheap computational cost. In the future, we plan to investigate network scaling in face recognition using NAS like EfficientNet.

References

[1] iqiyi-vid dataset. `http://challenge.ai.iqiyi.com/data-cluster`.

[2] J.-C. Chen, R. Ranjan, A. Kumar, C.-H. Chen, V. M. Patel, and R. Chellappa. An end-to-end system for unconstrained face verification with deep convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 118–126, 2015.

[3] 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.

[4] Y. Chen, H. Fang, B. Xu, Z. Yan, Y. Kalantidis, M. Rohrbach, S. Yan, and J. Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. *arXiv preprint arXiv:1904.05049*, 2019.

[5] A. R. Chowdhury, T.-Y. Lin, S. Maji, and E. Learned-Miller. One-to-many face recognition with bilinear cnns. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[6] J. Deng and J. Guo. Insightface. `https://github.com/deepinsight/insightface`.

[7] 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.

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

[9] 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.

[10] 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.

[11] Y. Feng, H. Wang, R. Hu, and D. T. Yi. Triplet distillation for deep face recognition. *arXiv preprint arXiv:1905.04457*, 2019.

[12] A. Gholami, K. Kwon, B. Wu, Z. Tai, X. Yue, P. Jin, S. Zhao, and K. Keutzer. Squeezenext: Hardware-aware neural network design. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1638–1647, 2018.

[13] S. Gong, Y. Shi, and A. K. Jain. Video face recognition: Component-wise feature aggregation network (c-fan). *arXiv preprint arXiv:1902.07327*, 2019.

[14] 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.

[15] 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.

[16] 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.

[17] G. Huang, S. Liu, L. Van der Maaten, and K. Q. Weinberger. Condensenet: An efficient densenet using learned group convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2752–2761, 2018.

[18] 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.

[19] 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.

[20] J. Karlekar, J. Feng, Z. S. Wong, and S. Pranata. Deep face recognition model compression via knowledge transfer and distillation. *arXiv preprint arXiv:1906.00619*, 2019.

[21] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882, 2016.

[22] 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.

[23] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[24] 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.

[25] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, page 7, 2016.

[26] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 116–131, 2018.

[27] 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.

[28] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *bmvc*, volume 1, page 6, 2015.

[29] 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.

[30] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[31] 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.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[33] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019.

[34] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

[35] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018.

[36] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.

[37] B. Wu, A. Wan, X. Yue, P. Jin, S. Zhao, N. Golmant, A. Gholaminejad, J. Gonzalez, and K. Keutzer. Shift: A zero flop, zero parameter alternative to spatial convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9127–9135, 2018.

[38] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4362–4371, 2017.

[39] 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.

[40] N. Zhu and X. Bai. Neural architecture search for deep face recognition. *arXiv preprint arXiv:1904.09523*, 2019.