MobiFace: A Lightweight Deep Learning Face Recognition on Mobile Devices

Chi Nhan Duong [1], Kha Gia Quach [1], Ngan Le [2], Nghia Nguyen [3], Khoa Luu [3]
[1] Computer Science and Software Engineering, Concordia University, Canada
[2] Electrical and Computer Engineering, Carnegie Mellon University, USA
[3] Computer Science and Computer Engineering, University of Arkansas, USA
[1]{dcnhan, kquach}@ieee.org, [2]thihoanl@andrew.cmu.edu,
[3]{nhnguyen, khoaluu}@uark.edu

Abstract— **Deep neural networks have been widely used in numerous computer vision applications, particularly in face recognition. However, deploying deep neural network face recognition on mobile devices is still limited since most high-accuracy deep models are both time and GPU consumption in the inference stage. Therefore, developing a lightweight deep neural network is one of the most promising solutions to deploy face recognition on mobile devices. Such the lightweight deep neural network requires efficient memory with small number of weights representation and low cost operators. In this paper a novel deep neural network named MobiFace, which is simple but effective, is proposed for productively deploying face recognition on mobile devices. The experimental results have shown that our lightweight MobiFace is able to achieve high performance with 99.7% on LFW database and 91.3% on large-scale challenging Megaface database. It is also eventually competitive against large-scale deep-networks face recognition while significant reducing computational time and memory consumption.**

I. Introduction

We have recently witnessed numerous computer vision applications with very high accurate performance, e.g. object detection [10], [48], [50], [9], [47], [49], [30], object classification [16], [7], [6], object segmentation [15], [33], [28], [29], [12], and modeling [8] etc. These major achievements have been accomplished thank to the popular Deep Convolutional Neural Networks (CNNs) methods. However, such high accurate performance CNNs usually require millions of parameters and several hundreds of layers to discriminate the classification spaces. For instance, Alex-Net [27] requires 61 million parameters, VGG-16 [42] requires 138 million parameters, ResNet-50 [17] requires 25 million parameters, DenseNet-190 ($k = 40$)[20] needs 40 million parameters. Although these aforementioned networks such as Alex-Net or VGG-16 are nowadays considered as not very deep models, they still cost approximately 200 MB and 500 MB memory size when implemented in Caffe framework, respectively. The large memory and powerful GPU resources are required to deploy in these methods to achieve high performance results. Therefore, such models are usually unable to deploy on power-hungry or mobile devices due to their model sizes and computational costs. To overcome these limitations while maintaining high performance, some recent developments are trying to compress the networks and known as compressed networks. Some well-known compressed networks such as Pruning [14], [13], [32], Depth-wise Convolution [18], [38], BinaryNets [22], [3], [36], [4], Mimic Networks [31], [44]. These networks can speed up the inference stage in CNNs without suffering accuracy loss in the fundamental tasks of image classification or object detection. However, the performance of these compressing methods hasn't been benchmarked on face recognition problems. Far apart from object detection and image classification, face recognition problems are usually required a considerable numbers of layers in the networks so that they are robust enough to present discriminated facial deep features across hundred thousands or millions of facial subjects. Indeed, these subjects have almost the same facial template with two eyes, a nose and a mouth.

In this paper, we introduce a novel *lightweight* but *high-performance* deep neural network for face recognition on mobile devices. Compare to the prior deep learning based face recognition methods, the contributions of our proposed MobiFace approach can be summarized as follows: Firstly, we improve the successful MobileNet framework [1] to lighter-weight and better deep network MobiNet model that is suitable for deploying on mobile devices

Secondly, the proposed MobiNet is then applied in face recognition and optimized within an end-to-end deep learning framework. Finally, we conduct the experiments and compare the proposed MobiNet against both mobile-based network and large-scale deep-network on face recognition task and on two state-of-the-art face recognition databases, i.e. Labeled Faces in the Wild (LFW) and large-scale challenging Megaface databases.

II. Related Works

Regarding lightweight deep network design, there are some well-known networks such as binarized networks, quantized networks, mimicked networks, designed compact modules and pruned networks. In this section, we would like to review the last two designs which are closed to our proposed network.

Designed compact modules. Integrating small modules or compact blocks, layers can abate the number of weights, help use less memory, and mitigate heavy computation cost for inference stage. The work of Andrew et al. [18] in Google, called MobileNet, proposed a depthwise separable convolution module instead of standard convolution to reduce a significant number of parameters. Depthwise separable convolution operation was first used in the thesis of Sifre In 2014 [41] and is applied to many networks [2], [18], [38]. In MobileNet [18], the spatial input convolves with a 3x3 spatial separate-channel filter to generate independent features, then a pointwise (i.e. 1x1) convolution operation combines these features to achieve new features. So this operation is considered as a replacement of standard convolution. MobileNet with only 4.2 million parameters and 569 million of mults-adds achieves promising results on the challenge image classification dataset of ImageNet [5] with the accuracy reaches to 70.6% whereas the performance of VGG-16 is 71.5% with 138 million parameters and 15300 million of mults-adds. Even reducing the number of parameters to 33x and number of mults-adds to 27x, MobileNet archives almost similar classification performance compare against VGG-16. To improve MobileNet performance on multiple tasks and benchmarks, Sandler et al. [38] proposed inverted residuals and linear bottlenecks, called MobileNet-V2. Inverted residuals are similar to residual bottleneck proposed in [16] but the intermediate features are expanded to a specific ratio w.r.t the number of input channels. Linear bottlenecks are blocks without ReLU layer in the end. MobileNet-V2 slightly improves the performance of MobileNet [18], achieves 72% accuracy on ImageNet [5] with only 3.4 million of parameters and takes 300 million of mults-add for computation. Although the depthwise separable convolution is proved efficient for designing network, networks [38], [18] still occupy a lot of memory and computational cost on iPhone or Android. To the best of our knowledge depthwise convolution has not been optimized to effectively run on CPU in most of deep learning frameworks such as Caffe [24], Pytorch [34], Tensorflow, etc. Also aiming at reducing the computational cost of MobileNet, fast downsampling approach is also employed in FD-Mobilenet [35]. Motivating from the structure of MobileNet-V2, MobileFaceNets [1] was proposed and adopted to face recognition problem. Similar to MobileNet-V2, residual bottleneck blocks are also considered as the building blocks of MobileFaceNets. Although sharing similar design strategy in the structure of MobileNet-V2, MobileFaceNets' architecture is smaller by replacing the global averaging pooling layer with the global depthwise convolution layer to weight pixels at different locations differently.

Pruned networks. Deep neural networks are over-parameterized and costly memory. Song Han et al. [14], [14] proposed a deep compression method to prune unimportant connections by absolute values; pruning achieve compression rate of 9x and 13x on AlexNet [27] and VGG-16 [42] respectively for ImageNet without suffering accuracy loss. Another idea from Liu et at. [32] is to slim networks using scaling factors in Batch Normalization layers [23] instead of absolute values of weights. These scaling factors are trained sparsely via L1-regularization technique [39]. Slimming networks [32] attain many good results in VGG-16 [42], DenseNet [20], and ResNet [16] even the accuracy is better than the original networks on datasets: CIFAR-10, CIFAR-100 [26]. However, for each pruned connections list of indices need to be stored to memory, leading to a very low progress for both training and testing.

III. Our Proposed MobiNet

This section starts with introducing network design strategies to construct a lightweight deep network. Then, by adopting these strategies, the architecture of MobiFace for face recognition on mobile devices is introduced. Thanks to the concise and precise deep network architecture, the proposed framework is efficient in terms of small computational cost and high accuracy in comparison against other deep networks on several large-scale face recognition databases.

A. Network Design Strategy

Bottleneck Residual block with the expansion layers. The use of Bottleneck Residual block is introduced in [37] where a block consists of three main transformation operators, i.e. two linear transformations and one non-linear per-channel transformation. There are three key factors of this type of block: (1) the non-linear transformation to learn complex mapping functions; (2) the layer expansion with increasing number of feature maps in the inner layers; and (3) shortcut connections to learn the residual. Formally, given an input $\mathbf{x}$ with the size of $h \times w \times k$, a bottleneck residual block can be represented as follows,

$$\mathscr{B}(\mathbf{x}) = [\mathscr{F}_1 \circ \mathscr{F}_2 \circ \mathscr{F}_3](\mathbf{x}) \tag{1}$$

where $\mathscr{F}_1 : \mathbb{R}^{w\times h\times k} \mapsto \mathbb{R}^{w\times h\times tk}$ and $\mathscr{F}_3 : \mathbb{R}^{w\times h\times k} \mapsto \mathbb{R}^{\frac{w}{s}\times\frac{h}{s}\times k_1}$ are the linear function represented by 1×1 convolution operator, and t denotes the expansion factor. $\mathscr{F}_2 : \mathbb{R}^{w\times h\times tk} \mapsto \mathbb{R}^{\frac{w}{s}\times\frac{h}{s}\times tk}$ is the non-linear mapping function which is a composition of three operators, i.e. ReLU, 3×3 depthwise convolution with stride s, and ReLU.

The residual learning connection is employed in a bottleneck block. This type of blocks is shown to have the capabilities of preventing manifold collapse during transformation and also increasing the expressiveness of the feature embedding [37].

Fast Downsampling. Under the limited computational resource of mobile devices, a compact network should *maximize the information transferred from the input image to output features* while *avoiding the high computational cost due to the large spatial dimensions* of feature maps. In the large-scale deep networks, the detail information flow is usually ensured by the *slow downsampling strategy*, i.e. the spatial dimensions are slowly reduced between blocks by downsampling operator. Consequently, the these networks maintain so many feature maps with large spatial size and result in a heavy-size network. On the other hand, under the limited computational budgets, a light-weight network adopting that slow downsampling may suffer both issues of weak feature embedding and high processing time.

In these cases, *fast downsampling strategy* can be considered as an efficient replacement of the slow downsampling

TABLE I

MODEL ARCHITECTURE FOR FACIAL FEATURE EMBEDDING. /2 MEANS THE OPERATOR HAS STRIDE OF 2. "BLOCK" AND "RBLOCK" INDICATE THE BOTTLENECK BLOCK AND RESIDUAL BOTTLENECK BLOCK, RESPECTIVELY.

Input	Operator
112 × 112 × 3	3 × 3 Conv, /2, 64
56 × 56 × 64	3 × 3 DWconv, 64
56 × 56 × 64	Block 1 × { 1 × 1 Conv, 128 3 × 3 DWconv, /2, 128 1 × 1 Conv, Linear, 64
28 × 28 × 64	RBlock 2 × { 1 × 1 Conv, 128 3 × 3 DWconv, 128 1 × 1 Conv, Linear, 64
28 × 28 × 64	Block 1 × { 1 × 1 Conv, 256 3 × 3 DWconv, /2, 256 1 × 1 Conv, Linear, 128
14 × 14 × 128	RBlock 3 × { 1 × 1 Conv, 256 3 × 3 DWconv, 256 1 × 1 Conv, Linear, 128
14 × 14 × 128	Block 1 × { 1 × 1 Conv, 512 3 × 3 DWconv, /2, 512 1 × 1 Conv, Linear, 256
7 × 7 × 256	RBlock 6 × { 1 × 1 Conv, 512 3 × 3 DWconv, 512 1 × 1 Conv, Linear, 256
7 × 7 × 256	1 × 1 Conv, 512
7 × 7 × 512	512-d FC

technique. In particular, in fast downsampling, the downsampling steps are consecutively applied in the very beginning stage of the feature embedding process to avoid large spatial dimension of the feature maps. Then in the later stage, more feature maps are added to support the information flow of the whole network. By this way, more complex mapping functions are learned to generate more details feature. Notice that, in this strategy, even more feature maps were added to the later feature, i.e. increase the number of channels, the computational cost is maintained to be low since the spatial dimensions of these feature maps are small.

B. MobiFace

In this section, we present a novel simple but efficient deep network for face recognition, named MobiFace. Given an input facial image with the size of $112 \times 112 \times 3$, this light-weight network aims at maximizing the information embedded in final feature vector while maintaining the low computational cost. Inspired by the strategies presented in the previous section, the Residual Bottleneck block with expansion layers is adopt as the building block of MobiFace. Table I represents the main architecture of MobiFace that consists of one 3×3 convolutional layer, one 3×3 depthwise separable convolutional layer, followed by a sequence of Bottleneck blocks and Residual Bottleneck blocks, one 1×1 convolutional layer, and a fully connected layer. The structures of Residual Bottleneck blocks and Bottleneck blocks are very similar except a shortcut is added to connect the input and the output of the 1×1 convolution layer. Moreover, the stride s is set to 2 in Bottleneck blocks while that parameter is set to 1 in every layers of Residual Bottleneck blocks.

Moreover, we adopt the fast downsampling strategy in our network architecture by quickly reducing the spatial dimensions of layers/blocks with the input size larger than 14×14. As one can easily see that given an input image with the size of $112 \times 112 \times 3$, the spatial dimension is reduced by half within the first two layers and become $8\times$ smaller after the other 7 bottleneck blocks. The expansion factor is kept to 2 whereas the number of channels is double after each Bottleneck block in later feature embedding stage.

A batch normalization together with a non-linear activation are applied after each convolutional layer except the one marked as "Linear". In our implementation, PReLU is used for the non-linear activation function due to its accuracy improvement over ReLU function. In the last layer of MobiFace, rather than employing the Global Average Pooling (GAP) layer as in previous approaches [19], [37], [1], we use the Fully Connected (FC) layer in the last stage of embedding process. Compared to GAP which treats very units in the last convolutional layer equally (*which is not very efficient since the information in the center pixel should play more important role than the one in the corner of the input*) , the FC layer can learn different weights to these units and gain the information embedded in the final feature vector.

IV. EXPERIMENTAL RESULTS

We first train the network using the cleaned training set of MS-Celeb-1M [11] including 3.8 million photos from 85K subjects. Then the trained network is evaluated on two common large-scale face verification benchmarks in unconstrained environments such as Labeled Faces in the Wild (LFW) [21], and Megaface [25] datasets. This training data has no overlapping with the testing data.

The databases that are used for training and testing are first described in next subsections. Then the comparisons between different models in terms of both accuracy and model sizes are represented. MobiFace can achieve very high performance, even competitive against other large-scale deep networks for face recognition.

A. Databases

MS-Celeb-1M [11] is introduced as a large-scale face dataset with 10 million photos of 100K celebrities. However, it also contains a large number of noisy image or wrong ID labels. To obtain a high-quality training data, the MS-Celeb-1M cleaned up the MS-Celeb-1M by computing the center feature of each subject and ranking their face images using the distance to identity center. The ones far from the center are automatically removed. Some manual checks are also employed. The refined MS-Celeb-1M consists of 3.8M photos from 85 identities.

Labeled Faces in the Wild (LFW) [21] is one of the common testing dataset for face verification. LFW consists 13,233 in-the-wild facial images of 5749 subjects collected from the web. The face variations include pose, expression and illuminations. According to the testing protocol of LFW,

TABLE II

Performance of Different face matching methods on LFW benchmark. * stands for our re-implementation.

Methods	# Training images	Model Size	Accuracy (%)
Google-FaceNet [40]	200M	30MB	99.63%
CosFace [43]	5M		99.73%
LightCNN [45]	4M	50MB	99.33%
MobilenetV1 [18] *	3.8M	112MB	99.50%
MobileFaceNet [1] *	3.8M	4MB	99.48%
MobiFace (Ours)	3.8M	9.3MB	**99.7%**

there are 6000 face pairs where half of them are positive pairs.

MegaFace [25] is one of largest publicly available testing dataset for face verification. This testing protocol is very challenging with million scale of distractors, i.e. subjects are not in the testing set. There are two main sets in Megaface, i.e. gallery and probe set. The gallery set is collected from Flickr photos and consists of more than 1 millions images from 690K identities. The probe set in Megaface are collected from two existing databases: Facescrub As the Facescrub probe set aims at the robustness of face recognition systems on large number of identity, this set includes 100K photos of 530 subjects. Meanwhile, the FG-NET probe set focuses on the robustness of the system against age changing, with 1002 images of 82 identities from 0 to 69 years old. In this paper, we evaluate the performance of our light-weight network on the Facescrub probe set.

B. Implementation details

In the preprocessing step, MTCNN method [46] is applied to detect all faces and their five landmark points, i.e. two eye centers, nose and two mouth corner, in both training and testing photo. Then, using the information from five landmark points, each face is aligned and cropped into a template with the size of $112 \times 112 \times 3$. This template is then normalized into $[-1, 1]$ by subtracting the mean pixel value, i.e. 127.5, and divided by 128.

During training stage, we adopt Stochastic Gradient Descent (SGD) optimizer with the batch size of 1024. The momentum parameter is set to 0.9. The learning rate is initialized to 0.1 and decreases by a factor of 10 periodically at 40K, 60K, and 80K iterations. The training stage is stopped at 100K iteration.

C. Face Verification accuracy

LFW benchmark. We first compare our MobiFace against many existing face recognition approaches including both large-scale deep models and small-scale one. Table II represented the performance of different matching methods on LFW benchmark. From these results, one can easily see that our MobiFace achieves 99.7% with the model size of only 9.3MB. With this performance, our MobiFace outperforms other small-size models and achieves competitive results to other large-scale deep models. MobiFace also outperforms most of other approaches in Table II.

TABLE III

Performance of Different face matching methods on the refined version of Megaface benchmark with one million distractors. All the models are train witn Large training dataset, i.e. > 0.5M. Model size is large when its size is greater than 20MB. * stands for our re-implementation.

Methods	Model Size	Accuracy (%)
MobilenetV1 [18] *	Large	92.65%
Google-FaceNet [40]	Large	86.47%
MobileFaceNet [1]*	Small	90.71%
MobiFace (Ours)	Small	**91.3%**

Megaface benchmark. We further validate the performance of our light-weight MobiFace on the challenging Megaface benchmark against millions of distractors. Table III illustrates the verification results of different methods on Megaface. The accuracy is reported on the True Accepted Rate (TAR) at the False Accepted Rate (FAR) of 10^{-6}. These results again emphasize the performance of our MobiFace when it outperforms the other light-weight MobileFaceNet model. Compared to other large-scale deep networks, our MobiFace has the advantages of both comparable performance to these models while maintaining low computational cost. Therefore, our MobiFace is easy to be deployed on mobile devices.

V. Conclusion

This paper has reviewed different lightweight deep network structures and approaches for mobile devices where the computational resource is very limited. Inspired by different network design strategies, this paper has further presented a novel simple but high-performance deep network for face recognition, named MobiFace. Experiments on two common large-scale face verification benchmarks with photo in unconstrained environment have shown the efficiency of our MobiFace in terms of both accuracy and small model size. Although the model is very small, its performance on both testing benchmarks is competitive against other large-scale deep face recognition network.

References

[1] S. Chen, Y. Liu, X. Gao, and Z. Han. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. *arXiv preprint arXiv:1804.07573*, 2018.

[2] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pages 1800–1807. IEEE Computer Society, 2017.

[3] M. Courbariaux and Y. Bengio. Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1. *CoRR*, abs/1602.02830, 2016.

[4] M. Courbariaux, Y. Bengio, and J. David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *NIPS*, pages 3123–3131, 2015.

[5] J. Deng, W. Dong, R. Socher, L. jia Li, K. Li, and L. Fei-fei. Imagenet: A large-scale hierarchical image database. In *In CVPR*, 2009.

[6] C. N. Duong, K. Luu, K. Quach, and T. Bui. Beyond principal components: Deep boltzmann machines for face modeling. In *CVPR*, 2015.

[7] C. N. Duong, K. Luu, K. Quach, and T. Bui. Longitudinal face modeling via temporal deep restricted boltzmann machines. In *CVPR*, 2016.

[8] C. N. Duong, K. Luu, K. Quach, and T. Bui. Deep appearance models: A deep boltzmann machine approach for face modeling. *Intl Journal of Computer Vision (IJCV)*, 2018.

[9] C. N. Duong, K. G. Quach, K. Luu, T. H. N. Le, and M. Savvides. Temporal non-volume preserving approach to facial age-progression and age-invariant face recognition. In *ICCV*, 2017.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014.

[11] 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.

[12] M. S. H. N. Le, R. Gummadi. Deep recurrent level set for segmenting brain tumors. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 646–653. Springer, 2018.

[13] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *CoRR*, abs/1510.00149, 2015.

[14] S. Han, J. Pool, J. Tran, and W. J. Dally. Learning both weights and connections for efficient neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1135–1143, Cambridge, MA, USA, 2015. MIT Press.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[17] 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.

[18] 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. *CoRR*, abs/1704.04861, 2017.

[19] 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.

[20] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 2261–2269, 2017.

[21] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

[22] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. Binarized neural networks. In *NIPS*, pages 4107–4115, 2016.

[23] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, volume 37, pages 448–456. JMLR.org, 2015.

[24] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*, pages 675–678. ACM, 2014.

[25] 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.

[26] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[28] H. N. Le, C. N. Duong, K. Luu, and M. Savvides. Deep contextual recurrent residual networks for scene labeling. In *Journal of Pattern Recognition*, 2018.

[29] H. N. Le, K. G. Quach, K. Luu, and M. Savvides. Reformulating level sets as deep recurrent neural network approach to semantic segmentation. In *Trans. on Image Processing (TIP)*, 2018.

[30] H. N. Le, C. Zhu, Y. Zheng, K. Luu, and M. Savvides. Robust hand detection in vehicles. In *Intl. Conf. on Pattern Recognition (ICPR)*, 2016.

[31] Q. Li, S. Jin, and J. Yan. Mimicking very efficient network for object detection. *2017 IEEE Conference on CVPR*, pages 7341–7349, 2017.

[32] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang. Learning efficient convolutional networks through network slimming. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2755–2763, 2017.

[33] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*.

[34] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

[35] Z. Qin, Z. Zhang, X. Chen, C. Wang, and Y. Peng. Fd-mobilenet: Improved mobilenet with a fast downsampling strategy. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1363–1367. IEEE, 2018.

[36] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV (4)*, volume 9908 of *Lecture Notes in Computer Science*, pages 525–542. Springer, 2016.

[37] 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.

[38] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018.

[39] M. W. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for l1 regularization: A comparative study and two new approaches. In *ECML*, 2007.

[40] 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.

[41] L. Sifre. Rigid-motion scattering for image classification, 2014.

[42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[43] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[44] Y. Wei, X. Pan, H. Qin, and J. Yan. Quantization mimic: Towards very tiny cnn for object detection. *CoRR*, abs/1805.02152, 2018.

[45] X. Wu, R. He, Z. Sun, and T. Tan. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 13(11):2884–2896, 2018.

[46] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[47] Y. Zheng, C. Zhu, K. Luu, H. N. Le, C. Bhagavatula, and M. Savvides. Towards a deep learning framework for unconstrained face detection. In *BTAS*, 2016.

[48] C. Zhu, Y. Ran, K. Luu, and M. Savvides. Seeing small faces from robust anchor's perspective. In *CVPR*, 2018.

[49] C. Zhu, Y. Zheng, K. Luu, H. N. Le, C. Bhagavatula, and M. Savvides. Weakly supervised facial analysis with dense hyper-column features. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2016.

[50] C. Zhu, Y. Zheng, K. Luu, and M. Savvides. Enhancing interior and exterior deep facial features for face detection in the wild. In *Intl Conf. on Automatic Face and Gesture Recognition (FG)*, 2018.