Improved Knowledge Distillation for Training Fast Low Resolution Face Recognition Model

Mengjiao Wang*, Rujie Liu
Fujitsu Research & Development Center Co., Ltd.
Beijing, China
{wangmengjiao, rjliu}@cn.fujitsu.com

Nada Hajime, Narishige Abe, Hidetsugu Uchida, Tomoaki Matsunami
Fujitsu Laboratories Ltd.
Kawasaki, Japan
{nada.hajime, abe.narishige, u.hidetsugu, t.matsunami}@fujitsu.com

Abstract

Low resolution (LR) face recognition (FR) is a challenging, yet common problem for FR task, especially for surveillance scenario. The issue addressed here is not just to build a LR-FR model, more importantly to make it run fast. Here, the knowledge distillation method is adopted for our task, where the teacher's knowledge can be 'distilled' into a small student model by guiding its training process. For LR-FR task, the original knowledge distillation scheme would update the teacher's weights first by tuning it using LR augmented train set, and then the student model is trained using same train set under updated teacher's guidance. The problem of this method is that the weights tuning of large teacher model is time-consuming, especially for large-scale dataset. In this paper, we proposed an improved scheme to enable us to avoid the teacher retraining and still be able to train the small model for LR-FR task. Here, different from the original scheme, the train sets for teacher and student model become different, where the train set for teacher model keeps unchanged and the one student is LR augmented. Therefore, it becomes unnecessary to update teacher model any more since the train set is the unchanged. Only the small student model needs to be trained under the original teacher's guidance. This can speed up the whole training process, especially for large-scale dataset. The different train sets for teacher and student will increase the data distribution discrepancy. To solve this problem, we constrained the multi-kernel maximum mean discrepancy between outputs to reduce this influence. Experimental results show our method can accelerate the training process by about 5 times, while preserving the accuracy. Our student model has same level with respect to state-of-art accuracy on LFW and SCFace. It can achieve $3\times$ acceleration comparing to teacher model and only takes 35ms to run on a CPU.

*coresponding author: wangmengjiao@cn.fujitsu.com

1. Introduction

Low resolution (LR) is a common problem for face recognition (FR) task, especially for surveillance video. In this task, the probe face image is often of low resolution (LR), while the gallery image is usually high resolution (HR). The issue we addressed here is how to train a fast LR-FR model. Among various model compression techniques [1, 2, 3, 4, 5, 6], we adopted the knowledge distillation (also known as teacher-student training) [5, 6] for our purpose. The idea is to force the small student model's output to be the same with large teacher model's output during training process. This scheme can be formulated as equation (1), where S represents student model, T represents teacher model, and x denotes the training sample.

$$S_{output}(x) = T_{output}(x) \tag{1}$$

Due to the lack of LR training data from surveillance scenario, the common way is to augment the existing train set by simulating LR image. If adopting the original knowledge distillation scheme for LR-FR task, it usually contains two steps: (1) use LR augmented train set to update the teacher's weights; (2) use same train set to train the student model under the new teacher's guidance. These two steps are illustrated as Figure.1.A. The problem of this scheme is the parameter tuning for the large teacher model is complex, which makes the whole process time-consuming, especially for large-scale train set. Also, there are some other cases that the teacher model is provided by some third party,

which means we don't have any information about the training details. This could make the teacher retraining more difficult, sometimes even impossible to reproduce.

Our goal is to investigate about how to avoid the cumbersome retraining of teacher model and still be able to train a fast student model for LR-FR task. We proposed an improved knowledge distillation scheme, where the teacher model still uses the original train set and the student uses the LR augmented train set. Since the teacher train set is unchanged, it's unnecessary to update the teacher model. We only need to perform a single step, i.e. training the fast student model under the original teacher's guidance. This scheme is illustrated in Figure.1.B. Our improved scheme can make it much faster to get a small student model for LR-FR task, since the training of light weighted student is much less complex. This scheme can be formulated as equation (2), where S ,T ,x and have same meaning with equation (1), and Δ represents the LR variance that is used for data augmentation.

$$S_{output}(x + \Delta) = T_{output}(x) \tag{2}$$

A problem come along with the improved scheme is that since the teacher and student models use different train sets, the distribution discrepancy between the two train sets may harm the accuracy. Therefore, we adopted the domain similarity metric multi kernel maximum mean discrepancy (MK-MMD) [7, 8] as our loss function to reduce the domain discrepancy and improve the performance.

Our contribution in this paper can be summarized as two aspects:

1. Proposed an improved knowledge distillation scheme to speed up the whole process of fast LR-FR model training. In our scheme, only the student's trainset is LR-augmented, while the teacher's input stays unchanged. This method can significantly accelerate the training process by avoiding the time-consuming weights updating of teacher model.

2. Constrain the MK-MMD metric to reduce the distribution discrepancy introduced by the different trainsets of teacher and student.

2. Method

In this part, first we will introduce the details about original knowledge distillation scheme, in which the selection of loss function, student model network architecture, and useful tricks will be explained in detail. Then the improved knowledge distillation scheme is introduced, including the details about LR simulated train set preparation, MK-MMD loss integration, etc.

2.1. Review of original knowledge distillation scheme

In knowledge distillation scheme, the output of the small student model is forced to be equal with teacher model's output. In this way, we can transfer the representation capability of teacher model to the student model. Here, we use combination of different loss function to implement this idea. The loss functions include: soft loss; hard loss; feature L2 loss.

Let's denote the final score output as Z , the soft label for teacher model T can be defined as $X_T^\tau = softmax(z_T/\tau)$ where τ is the temperature parameter. Similarly, the soft label for student network S is $X_S^\tau = softmax(z_S/\tau)$. The soft loss is the cross entropy between X_T^τ and X_S^τ:

$$L_{soft} = H(X_T^\tau, X_S^\tau) \tag{3}$$

The hard loss is the cross entropy between unsoften class probability X_S and ground truth y:

$$L_{hard} = H(X_S, y) \tag{4}$$

Here, $H(.)$ represents for cross entropy.

For the hint learning, we used the feature layer as hint to train the student model. The hint loss is actually feature L2 loss:

$$L_{Feature} = \|F_S - F_T\| \tag{5}$$

F_S and F_T are the features from student and teacher.

Among the three losses, the soft loss can transfer the knowledge from teacher to student by using soft label, the hard loss can make the student develop its own classification ability, and the hint learning can boost the performance and accelerate the convergence according to [6]. Here the feature is normalized before calculating L2 loss. This will enhance the overall performance according to [9]. The loss combination can be formulated as equation (6), where $L_{Feature\_Norm}$ denotes the L2 loss for normalized feature and λ_1, λ_2 , and λ_3 represent the weights for hard, soft and normalized feature loss.

$$L_{overall} = \lambda_1 L_{Hard} + \lambda_2 L_{Soft} + \lambda_3 L_{Feature\_Norm} \tag{6}$$

In this paper, our teacher model is a 64-layer ResNet [10] model. For the choice of network architecture for student training, several light weighted network architecture are available, including SqueezeNet [11], MobileNet [12], and ShuffleNet [13], state-of-art architecture model, including DenseNet [14], and Inception-ResNet [15], and thinner/deeper model. In [9], it's concluded that the thinner/deeper model will generate best performance when the student network has the similar network architecture but with less channels, and/or more depth. In our work, we adopted a 36-layer thinner ResNet network as our student model. The detailed network architecture will be demonstrated in the following part.

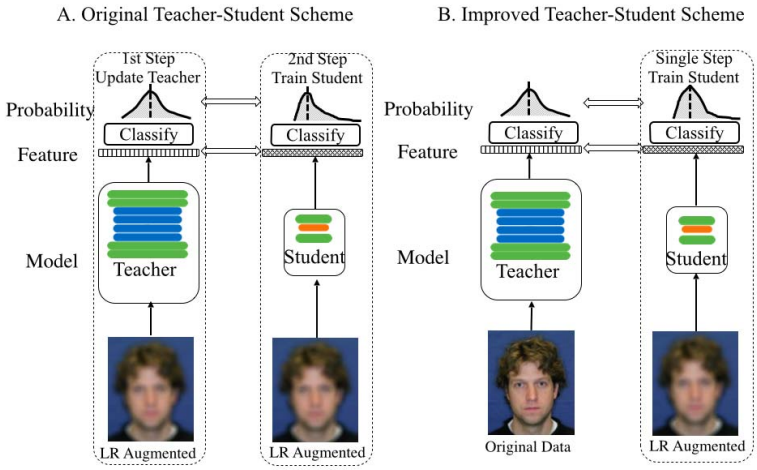

Figure 1. Comparison between original and our improved knowledge distillation schemes for training fast low resolution (LR) face recognition (FR) model. The original scheme is shown as 1.A, where the train datasets for teacher and student models need to be same. For training with LR FR model , original scheme adopts two steps: 1st step: update teacher using LR augmented train dataset; 2nd step: train student with same LR augmented trainset. Our improved scheme is shown as 1.B, where the updating of teacher model is avoided. It only have a single step: train student with LR augmented data, where teacher model's train dataset is unchanged.

2.2. Improved knowledge distillation scheme for LR-FR task

To augment the train set using LR variance, we can add Gaussian blur to the original train set. The Gaussian blur can be applied by convolve each pixel with a Gaussian filter. The size of the Gaussian kernel determines the downscale ratio. For fair comparison with other works, the CASIA Webface dataset [16] is used as training set. MTCNN [17] is used to detect and align the face region as 112×96 image. Among these face images, 40 percent of the training set are randomly selected for downscaling as LR samples. The size of the LR samples includes 8×8, 12×12, 16×16, 20×20 and 30×30.

In our improved knowledge distillation scheme, the train set for teacher model is unchanged, while the train set for student model is LR-augmented. This scheme may increase distribution discrepancy between teacher and student's train sets. In our work, we adopted the multi kernel maximum mean discrepancy (MK-MMD) [7, 18, 8] as our loss function to reduce the dataset discrepancy. MMD is widely used as a distribution distance to measure the discrepancy between two domains. It compares the distributions in the Reproducing Kernel Hilbert Space (RKHS) [18]. The equation for MMD can be formulated as:

$$L_{MMD}(x,y) = \| \frac{1}{N} \sum_{i=1}^{N} \phi(x^i) - \frac{1}{N} \sum_{j=1}^{N} \phi(y^j) \| \quad (7)$$

In the equation (7), $\phi(.)$ is an explicit mapping function. x^i and y^j represent two samples independently drawn from distributions of teacher and student's training datasets. By expanding equation (7), the equation can be reformulated as:

$$\begin{aligned} L_{MMD}(x,y) = & \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} k(x^i, x^{i'}) \\ & + \frac{1}{M^2} \sum_{j=1}^{N} \sum_{j'=1}^{N} k(y^j, y^{j'}) - \frac{2}{MN} \sum_{i=1}^{N} \sum_{j=1}^{N} k(x^i, y^j) \end{aligned} \quad (8)$$

From equation (8), we can see that MMD use kernel method $k(.,.)$ to project the sample vectors into higher dimension. Here, we use the Gaussian kernel $k(x,y) = exp(-\frac{\|x-y\|^2}{2\sigma^2})$. The σ^2 is set as the mean of squared distance of the pairs. In MK-MMD, we consider five Gaussian kernels by setting Gaussian distribution variance as $\sigma^2 \times (1, 2^1, 2^2, 2^3, 2^4)$ respectively [7].

To integrate the MK-MMD metric into the overall loss function, we replace the normalized feature L2 loss $L_{Feature\_Norm}$ in equation (6) with the normalized feature MK-MMD. Here, the x and y in equation (8) represents the feature extracted by teacher and student model respectively. In our method, instead of using the original x and y, the features used in equation (8) are normalized as $\frac{x}{\|x\|}$ and $\frac{y}{\|y\|}$ to make them have the same scale. We also try to apply MK-MMD for the soft label, but it slightly hurts the accuracy. Therefore, we only used MK-MMD for the normalized feature. The new overall loss function can be formulated as equation (9), where $L^{MK-MMD}_{Feature\_Norm}$ denotes the MK-MMD for normalized feature and $\bar{\lambda}_1$, $\bar{\lambda}_2$, and $\bar{\lambda}_3$ represent the weights for hard, soft, and normalized feature MK-MMD loss.

Conv1.x	Conv2.x	Conv3.x	Conv4.x	FC
$[3\times3, 64]\times1, S2$ $\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times3$	$[3\times3, 128]\times1, S2$ $\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times8$	$[3\times3, 256]\times1, S2$ $\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times16$	$[3\times3, 512]\times1, S2$ $\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times3$	512

Table 1. Network Configuration of Teacher Model. Here, [3×3, 128]×4, $S2$ denotes 4 cascaded convolution layers with 128 filters of size 3×3, stride is set as 2. The default stride is set as 1. 'FC' is the fully connected layer.

Conv1.x	Conv2.x	Conv3.x	Conv4.x	FC
$[3\times3, 18]\times1, S2$ $\begin{bmatrix}3\times3,18\\3\times3,18\end{bmatrix}\times2$	$[3\times3, 36]\times1, S2$ $\begin{bmatrix}3\times3,36\\3\times3,36\end{bmatrix}\times4$	$[3\times3, 72]\times1, S2$ $\begin{bmatrix}3\times3,72\\3\times3,72\end{bmatrix}\times8$	$[3\times3, 144]\times1, S2$ $\begin{bmatrix}3\times3,144\\3\times3,144\end{bmatrix}\times2$	512

Table 2. Network Configuration of Student Model. Here, [3×3, 128]×4, $S2$ denotes 4 cascaded convolution layers with 128 filters of size 3×3, stride is set as 2. The default stride is set as 1. 'FC' is the fully connected layer.

$$\bar{L}_{overall} = \bar{\lambda_1} L_{Hard} + \bar{\lambda_2} L_{Soft} + \bar{\lambda_3} L^{MK-MMD}_{Feature\_Norm} \quad (9)$$

3. Experiments and Result

3.1. Experiment Setting

Here, the test sets include LFW [19] and SCFace [20]. The LFW dataset is a widely used test set for face recognition engine. The faces in LFW are detected using MTCNN and aligned to 112×96 resolution. To simulate the setting of LR-FR, we modified the original LFW dataset, where the gallery image remains unchanged and the probe image is preprocessed to low resolution image by convolving with Gaussian kernels. The probe image resolution can be 8×8, 12×12, 16×16, 20×20, and the original 112×96. This new dataset is referred as low-resolution LFW (LR-LFW).

SCFace is a widely-used benchmark for evaluation of LR face recognition engine [21]. The SCFace database contains images of 130 subjects taken in uncontrolled indoor environment using five video surveillance cameras of various qualities. For each subject, there are 15 images in total taken at three different distances, 4.20m, 2.60m, and 1.00m, by surveillance cameras, and one frontal mugshot image taken by a digital camera. Here, the frontal mugshot images are used as gallery images, and images taken by surveillance cameras at different distances are used as probe image. The setting of detection and alignment are the same with LFW data processing.

3.2. Deep Neural Network Architecture Configuration

In our experiments, we used a pre-trained 64-layer ResNet network trained with Sphereface loss [22] as teacher model. The training data is the CASIA Webface dataset [16]. The teacher's network architecture is the same with the description in [22]. Here, the student model used a 36-layer thinner network. It can achieve 3× acceleration rate comparing to the teacher model. The network architecture for teacher and student models are shown as Table 1 and 2 respectively. In Table 1 and 2, Conv1.x, Conv2.x, Conv3.x and Conv4.x denote multiple convolution layers and residual units that are shown in double-column brackets. E.g., [3×3, 128]×4, $S2$ denotes 4 cascaded convolution layers with 128 filters of size 3×3, stride is set as 2. The default stride is set as 1. 'FC' is the fully connected layer.

3.3. Evaluation of improved knowledge distillation scheme

In this part, firstly, we compare the whole train time for the original and improved schemes, and then the performance of MK-MMD is evaluated. Finally, the method is compared with other state-of-art methods.

3.3.1 Evaluation for different knowledge distillation schemes

In this part, we conducted experiments to evaluate the whole training time and performance of LFW and SCFace of two different knowledge distillation schemes:

KD: Original knowledge distillation scheme, containing 2 steps. Step1, use LR augmented data to update teacher's weights; Step2, use same train set to train student model under updated teacher's guidance.

KD$_{im}$: Improved knowledge distillation scheme, containing a single step: using LR augmented train set to train student model under original teacher model's guidance.

In these two different strategies, **KD** adopted the original knowledge distillation scheme. **KD**$_{im}$ is our improved knowledge distillation scheme where only the student model is trained using LR augmented train set under the original teacher's guidance. The CASIA-webface dataset is used here for training. Here, we use 4 Nvidia Titan Xp GPUs with batchsize as 512. Caffe is adopted for training and the process will last for 20 epochs. The training for teacher model will roughly take 2 days to finish while

the student model training only take 0.5 days since it's much smaller. Therefore, the two steps in **KD** will take about 2.5 days in total to accomplish, while the **KD**$_{im}$ scheme's single step procedure only take 0.5 days. Since the improved scheme (**KD**$_{im}$) avoids the time consuming teacher retraining, the whole training process can be speeded up by about 5 times. The training time of these two different schemes are shown in Table 3.

For the large scale train set, the reduction of training time will be much more important. For example, if the DeepGlint [23] train set is adopted, which contains 181K ids and 6.75M images, the training of **KD** scheme may take more than a week by using same training parameter and hardware setting. By using **KD**$_{im}$ scheme, we can avoid the time consuming teacher's weights tuning and still be able to get the fast LR-FR model in about 1.5 days.

Train Scheme	Training Time
KD	2.5 days
KD$_{im}$	0.5 days

Table 3. Training time of different training strategies. 'KD' represents the original knowledge distillation scheme. 'KD$_{im}$' represents our improved knowledge distillation scheme.

Besides the acceleration of training time, the performance of these two knowledge distillation schemes are also compared on LR-LFW and SCFace dataset. The results are shown in Table 4 and Table 5. From the table, we can see that the student models trained with **KD**$_{im}$ scheme and **KD** scheme have same-level performance. Therefore, we can draw a conclusion that the student model with our improved knowledge distillation scheme can preserve the accuracy while accelerate the training process by 5 times.

In Table 4 and Table 5, the teacher model is only tested on the 112×96 LFW and d=1.0m SCFace images. These two image sets can be considered as high resolution (HR) images. The reason is that the teacher model used here hasn't been retrained using LR augmented dataset. Therefore, it's unfair to evaluate the accuracy on LR images.

PS	8×8	12×12	16×16	20×20	112×96
T	-	-	-	-	99.42
S$_{KD}$	93.95	95.08	97.00	97.10	99.17
S$_{KDim}$	94.05	95.20	96.74	97.13	99.03

Table 4. Accuracy of different models on LR-LFW dataset. Here, 'PS' is the abbreviation for 'probe size'. 'T' represents teacher model. 'S$_{KD}$' represents student model trained with original knowledge distillation scheme. 'S$_{KDim}$' represents student model trained with our improved knowledge distillation scheme.

Distance	$d = 4.2m$	$d = 2.6m$	$d = 1.0m$
T	-	-	99.13
S$_{KD}$	73.88	93.50	98.34
S$_{KDim}$	73.20	93.95	98.03

Table 5. Face Recognition rates of different models on SCFace dataset. Here, 'T' represents teacher model. 'S$_{KD}$' represents student model trained with original knowledge distillation scheme. 'S$_{KDim}$' represents student model trained with our improved knowledge distillation scheme.

3.3.2 Performance of MK-MMD loss

In this part, we mainly focused on the evaluation of MK-MMD loss function in our improved knowledge distillation scheme. We compared two experiments:

L2 loss: Use equation (6) as loss function in improved knowledge distillation scheme, where the feature L2 loss is adopted instead of MK-MMD loss.

MK-MMD loss: Use equation (9) as loss function in improved knowledge distillation scheme, where the feature MK-MMD loss is adopted.

The results on two datasets (LR-LFW, SCFace) are shown are shown in Table 6 and Table 7.

PS	8×8	12×12	16×16	20×20	112×96
S$_L$	90.36	91.75	94.66	96.83	99.00
S$_M$	94.05	95.20	96.74	97.13	99.03

Table 6. Evaluation on LR-LFW for student model trained with different loss combination. 'PS' is the abbreviation for 'probe size'. 'S$_L$' represents the student model trained with L2 loss combination. 'S$_M$' represents the student model trained with MK-MMD loss combination.

Distance	$d = 4.2m$	$d = 2.6m$	$d = 1.0m$
S$_L$	69.37	91.54	97.98
S$_M$	73.20	93.95	98.03

Table 7. Evaluation of SCFace face recognition rates using student model trained with different losses. 'S$_L$' represents the student model trained with L2 loss combination. S$_M$ represents the student model trained with MK-MMD loss combination.

From Table 6 and Table 7, we can see that results of MK-MMD loss constantly outperform L2 loss on both test sets, especially on smaller size probe image scenarios, such as 8×8, 12×12 LR-LFW images and $d = 4.2m$ SCFace images. The reason is that when using original feature L2 loss, the data distribution discrepancy between teacher's unchanged train set and student's LR-augmented train set can harm the performance due to the weak constraint of l2 loss function. This problem will become more severe when the probe image size becomes smaller and smaller. The influ-

ence of this problem can be reduced by adopting feature MK-MMD loss function, which can be seen as a combination of kernel trick and L2 loss and will exert stronger constraint on features during the training process.

3.3.3 Performance comparison on SCFace

In this part, results on SCFace are compared with three state-of-art methods, including Deep Coupled Resnet model (DCR) method [24] , multidimensional scaling (MDS) [25], and discriminative multidimensional scaling (DMDS) [26], (Table 8). The inference time of best performance methods, including DCR and our student model are tested on an Intel Core 2.5GHz CPU to simulate embedded systems' hardware performance. The inference time is shown in Table 9. From the results, we can safely draw a conclusion that our model can achieve comparable, sometime even better performance while be able to reduce the inference time to 35ms.

Distance	$d = 4.2m$	$d = 2.6m$	$d = 1.0m$
Teacher	-	-	99.13
S_{KDim}	73.20	93.95	98.03
DCR[24]	73.30	93.50	98.00
DMDS[26]	62.70	70.70	65.50
MDS[25]	60.30	66.00	69.50

Table 8. Face recognition rates of different models at different distances on SCFace. 'Teacher' represents the 64-layer resnet teacher model. 'S_{KDim}' represents the student model trained with our improved knowledge distillation scheme.

Model	Inference Time
S_{KDim}	$35ms$
DCR[24]	$132ms$

Table 9. Inference time of our model and DCR model on CPU. S_{KDim} represents the student model trained with our improved knowledge distillation scheme.

The reason behind the better performance may be concluded as two aspects:

(1) Better teacher model: the 64-layer teacher model has higher performance on HR face images, i.e. $d = 1.0m$ SCFace, comparing with DCR and other approaches (Table 8). The larger scale network structure and superior Sphereface loss function are the major reasons for teacher's better performance. Thanks to the improved knowledge distillation scheme, our student model's performance only drops slightly comparing to teacher model on the high resolution image sets.

(2) The improved knowledge distillation scheme only use LR augmented train set for student model, this forces the student model's output to be the same with teacher's output regardless of the LR variance. This can transfer the teacher's better representation capability to student model no matter the input is LR or HR, which increases student model's performance w.r.t. LR image.

4. Conclusion

In this paper, we proposed an improved knowledge distillation scheme for fast LR-FR model training. To avoid the time-consuming training process of teacher model, we keep the teacher model's train set unchanged, while only adding LR augmentation to the student model's train set. This can allow us to avoid the updating of teacher model's weights and still be able to train a LR-FR student model, which will reduce the time cost of the whole training process. Only adding LR augmentation to the student model's train set will increase the distribution discrepancy between teacher and student's training inputs. This discrepancy can be reduced by minimizing MK-MMD loss function. The results show that our method can reduce the training time by about 5 times while preserving the student model's accuracy. Our student model can achieve $3\times$ acceleration comparing to teacher model and only takes 35ms to run on a CPU. The improved scheme can also be generalized to other data variance, such as illumination, pose, etc.

References

[1] Wenlin Chen, James T. Wilson, Stephen Tyree, Kilian Q. Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. *CoRR, abs/1504.04788*, 2015.

[2] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115*, 2014.

[3] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *CoRR, abs/1510.00149*, 2015.

[4] Alexander Novikov, Dmitry Podoprikhin, Anton Osokin, and Dmitry P. Vetrov. Tensorizing neural networks. *In Advances in Neural Information Processing Systems, NIPS*, 2015.

[5] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *In Deep Learning and Representation Learning Workshop, NIPS*, 2014.

[6] Ping Luo, Zhenyao Zhu, Ziwei Liu, Xiaogang Wang, and Xiaoou Tang. Face model compression by distilling knowledge from neurons. *30th AAAI Conference on Artificial Intelligence*, pages 3560—-3566, 2016.

[7] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[8] Zimeng Luo, Jiani Hu, Weihong Deng, and Haifeng Shen. Deep unsupervised domain adaptation for face recognition.

2018 13th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2018), 2018.

[9] Mengjiao Wang, Rujie Liu, Narishige Abe, Hidetsugu Uchida, Tomoaki Matsunami, and Shigefumi Yamada. Discover the effective strategy for face recognition model compression by improved teacher-student training. *In 2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2416–2420, 2018.

[10] Kaiming He, Xiangyun Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *In Computer Vision and Pattern Recognition, CVPR*, 2016.

[11] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡1mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[12] Andrew G. Howard, Menglou Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[13] Xiangyun Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *arXiv preprint arXiv: 1707.01083*, 2017.

[14] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *In Computer Vision and Pattern Recognition, CVPR*, 2017.

[15] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR, abs/1602.07261*, 2016.

[16] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z. Li. Learning face representation from scratch. *CoRR, abs/1411.7923*, 2014.

[17] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *In IEEE Signal Processing Letters, 23(10)*, pages 1499–=1503, 2016.

[18] Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017.

[19] Gary B. Huang, Marwan Ramesh, Tarama Berg, and Eric Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *In University of Massachusetts, Amherst, Technical Report*, 2007.

[20] Mislav Grgic, Kresimir Delac, and Sonja Grgic. Scface—surveillance cameras face database. *Multimedia Tools Appl.*, pages 863—-879, 2011.

[21] Zhiwu Huang, Shiguang Shan, Ruping Wang, Haihong Zhang, Shihong Lao, Alifu Kuerban, and Xilin Chen. A benchmark and comparative study of video-based face recognition on scface face database. *IEEE Transactions on Image Processing*, 2015.

[22] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. *CVPR*, 2017.

[23] http://trillionpairs.deepglint.com/overview.

[24] Ze Lu, Xudong Jiang, and Alex Kot. Deep coupled resnet for low-resolution face recognition. *IEEE Signal Process. Lett.,*, pages 526—-530, 2018.

[25] Sivaram P. Mudunuri and Soma Biswas. Low resolution face recognition across variations in pose and illumination. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1034—-1040, 2016.

[26] Fuwei Yang, Wenming Yang, Riqiang Gao, and Qingming Liao. Discriminative multidimensional scaling for low-resolution face recognition. *IEEE Signal Process. Lett.*, pages 388—-392, 2018.