Teacher Guided Neural Architecture Search for Face Recognition

Xiaobo Wang[1,2]
[1]Sangfor Technologies Inc., Shenzhen, China
[2]CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
wangxiaobo2015cbsr@gmail.com

Abstract

Knowledge distillation is an effective tool to compress large pre-trained convolutional neural networks (CNNs) or their ensembles into models applicable to mobile and embedded devices. However, with expected flops or latency, existing methods are hand-crafted heuristics. They propose to predefine the target student network for knowledge distillation, which may be sub-optimal because it requires much effort to explore a powerful student from the large design space. In this paper, we develop a novel teacher guided neural architecture search method to directly search the student network with flexible channel and layer sizes. Specifically, we define the search space as the number of the channels/layers, which is sampled based on the probability distribution and is learned by minimizing the search objective of the student network. The maximum probability for the size in each distribution serves as the final searched width and depth of the target student network. Extensive experiments on a variety of face recognition benchmarks have demonstrated the superiority of our method over the state-of-the-art alternatives.

Introduction

Convolutional neural networks have gained impressive success in the recent advanced face recognition systems (Li et al. 2016; Zhao et al. 2017, 2018b,a, 2019; Wang et al. 2019b). However, the performance advantages are often driven at the cost of training and deploying resource-intensive networks with millions of parameters. As face recognition shifts toward mobile and embedded devices, the computational cost of large CNNs prevents them from being deployed to these devices. It motivates research of developing compact yet still discriminative models. Several directions such as model pruning, model quantization and knowledge distillation have been suggested to make the model smaller and cost-efficient. Among them, knowledge distillation is being actively investigated. For a pre-given larger network (teacher), the distillation process aims to learn a compact network (student) by utilizing the knowledge of teacher as supervision. The key to success in knowledge distillation is the designed student network and the exploited dark knowledge. Unlike other compression methods, it can downsize a network regardless of the structural difference between teacher and student networks.

For face recognition knowledge distillation, there have been several attempts (Wang, Lan, and Zhang 2017; Luo et al. 2016; Karlekar, Feng, and Pranata 2019; Ge et al. 2018; Feng et al. 2019; Peng et al. 2019; Wang et al. 2019a, 2020a) in literatures to distil large CNNs, so as to make their deployments easier. Hinton *et al.* (Hinton, Vinyals, and Dean. 2015) propose the first knowledge distillation based on the soften probability consistency, where a temperature parameter is introduced in the softmax function to disclose the similarity structure of data. Romero *et al.* (Romero, Ballas, and Kahou. 2014) used the hidden layer of a teacher network as a hint for a student network to improve the performance. Wang *et al.* (Wang, Lan, and Zhang 2017) use both the knowledge of soften probabilities and one-hot labels for face recognition and alignment. Luo *et al.* (Luo et al. 2016) propose a neuron selection method by leveraging the essential characteristics (domain knowledge) of the learned face representation. Karlekar *et al.* (Karlekar, Feng, and Pranata 2019) simultaneously exploit one-hot labels and feature vectors for the knowledge transfer between different face resolutions. Ge *et al.* (Ge et al. 2018) develop a selective knowledge distillation, which selectively distils the most informative facial features by solving a sparse graph optimization problem. Feng *et al.* (Feng et al. 2019) exploit the capability of a teacher model to transfer the similarity information to a small model by adaptively varying the margin between positive and negative pairs. Wang *et al.* (Wang et al. 2019a) propose an improved knowledge distillation scheme, where the teacher model uses the original training set and the student uses the low-resolution augmented training set. Peng *et al.* (Peng et al. 2019) use the knowledge of probability consistency to transfer not only the instance-level information, but also the correlation between instances. While these methods improve performance over the typical directly training the student network, they still come with a common limitation: the pre-defined student network is hand-crafted heuristic, which is sub-optimal because it requires much effort from experts to explore a suitable one from the large design space.

In this paper, we propose a novel teacher guided neural architecture search method from a probability optimization perspective. Based on the investigation of several popular knowledge used for distillation, we design a simple but very

effective search objective, *i.e.*, using feature fitting as a criterion for neural architecture search. For simplicity, we define the search space as the candidates of width and depth of the student network and formulate channels/layers as a parameterized probability distribution for sampling. To sum up, the main contributions of this paper can be summarized as:

- We investigate the effect of several knowledge for distilling face recognition models and come to the conclusion that the feature knowledge is more flexible and powerful than others. To this end, we utilize the feature fitting as our search objective for searching a suitable student network.
- We propose a novel teacher guided neural architecture search (TNAS) framework, which defines the search space as the width and depth of the student network and attaches different candidates of channels/layers with a learnable probability.
- We conduct extensive experiments on the face recognition benchmarks of LFW (Huang, Ramesh, and Miller. 2007), SLLFW (Deng et al. 2017), CALFW (Zheng et al. 2017), CPLFW (Zheng et al. 2018), AgeDB (Moschoglou et al. 2017), CFP (Sengupta et al. 2016), RFW (Wang et al. 2018c), MegaFace (Kemelmacher-Shlizerman et al. 2016) and Trillion-Pairs[1], which have verified the superiority of our approach over the state-of-the-arts.

Related Works

Face Recognition. Face recognition is an essential open-set metric learning problem, which is different from the closed-set image classification. Specifically, rather than using the traditional softmax loss, face recognition is usually supervised by margin-based softmax losses (Liu et al. 2017; Liang et al. 2017; Wang et al. 2018b,e; Deng et al. 2019; Wang et al. 2019c, 2020a,b), metric learning losses (Schroff, Kalenichenko, and Philbin 2015) or both (Sun et al. 2014). To achieve better performance, large CNNs like SEResNet (Deng et al. 2019) or AttentionNet (Wang et al. 2018d, 2019c) are usually employed, which makes them hard to deploy on mobile and embedded devices. Some works (Chen et al. 2018; Wu, He, and Tan 2018) start to design small networks for training, but the balance between the expected flops and the performance is usually unsatisfactory, which motivates us to use the knowledge distillation technique for face recognition model compression.

Knowledge Distillation. Many studies have been conducted since Hinton *et al.* (Hinton, Vinyals, and Dean. 2015) proposed the first knowledge distillation based on the soften class probabilities. Romerot *et al.* (Romero, Ballas, and Kahou. 2014) used the hidden layer response of a teacher network as a hint for a student network to improve knowledge distillation. Luo *et al.* (Luo et al. 2016) resorted to the top hidden layer as the knowledge and used the attributes to select the important neurons. Some studies (Li, Jin, and Yan 2017; Wang, Lan, and Zhang 2017; Chen, Wang, and Zhang 2018; Aguinaldo et al. 2019) extended knowledge distillation to other applications. Knowledge distillation has been studied in various directions, but the target student network is usually pre-defined, which may be sub-optimal because it requires much effort to explore a powerful one.

Neural Architecture Search. Neural architecture search (NAS) is a technique to turn the pre-defined architecture structures into a learning procedure. Instead of optimizing the network topology as prevailing methods (Liu and Simonyan 2018; Chen et al. 2019; Tan et al. 2019; Bashivan, Tensen, and DiCarlo 2019; Liu et al. 2020) do, we empirically observe that the network size is usually more crucial for face recognition knowledge distillation. For exploiting the depth and width of networks, several methods (Chen, Goodfellow, and Shlens 2015; Gordon et al. 2018; Cai et al. 2018; Dong and Yang 2019) have been proposed. Chen *et al.* (Chen, Goodfellow, and Shlens 2015) manually widen and deepen a network, and proposed Net2Net to initialize the lager network. Gordon *et al.* (Gordon et al. 2018) proposed a heuristic strategy to find a suitable width of networks by alternating between shrinking and expanding. Cai *et al.* (Cai et al. 2018) utilized a RL agent to grow the depth and width of CNNs. Dong *et al.* (Dong and Yang 2019) developed a differentiable approach from probability perspective to shrink CNNs. However, for face recognition knowledge distillation, they rarely involve and can not be directly extended to this case.

Methodology

For neural architecture search, the key mainly comes from two aspects: *i.e.*, search objective and search space. To this end, we first introduce the background of knowledge distillation and design the search objective to evaluate the searched student during training. We then clarify the search space of our teacher guided neural architecture search framework and show the details of its searching process.

Knowledge Distillation

In face recognition knowledge distillation, the common dilemma is that we only have a teacher model at hand and do not know how it was trained (including training sets, loss functions and training strategies of teacher *etc*). But the task is to distil a student network, which is with satisfactory performance as well as can be applicable to mobile and embedded devices. As a result, we have the following cases:

One-hot Labels. If the training set of student network is well-labelled, we can directly train the target student network with one-hot labels. Without loss of generality, we use the AM-Softmax loss (Wang et al. 2018b,a) as supervision to train the student. Obviously, this manner does not utilize the knowledge of teacher that hinders its performance.

Probability Knowledge Distillation (PKD). Let's denote the final softmax output as z, the soft label for teacher model T can be defined as $P_{\mathrm{T}}^{\tau} = (z_{\mathrm{T}}/\tau)$, where τ is the temperature parameter. Similarly, the soft label for student network S is $P_{\mathrm{S}}^{\tau} = (z_{\mathrm{S}}/\tau)$. Prevailing knowledge distillation approaches usually exploit the popular probability knowledge distillation as follows:

$$\mathcal{L}_{\mathrm{PKD}} := \mathcal{L}(P_{\mathrm{T}}^{\tau}, P_{\mathrm{S}}^{\tau}) = \mathcal{L}((z_{\mathrm{T}}/\tau), (z_{\mathrm{S}}/\tau)) \qquad (1)$$

where $\mathcal{L}$ is the cross entropy loss between P_{T}^{τ} and P_{S}^{τ}. However, the formulation of PKD is limited by softmax-based

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

losses and requires that the training classes of teacher and student should be same, which is not flexible because we usually only have a teacher model at hand but do not know how it was trained.

Feature Knowledge Distillation (FKD). In face recognition, we can also use the feature layer as hint to train the student network. The feature knowledge distillation can be formulated as follows:

$$\mathcal{L}_{\mathrm{FKD}} := \mathcal{H}(F_{\mathrm{S}}, F_{\mathrm{T}}) = ||F_{\mathrm{S}} - F_{\mathrm{T}}||, \tag{2}$$

where $\mathcal{H}$ is the L2 loss, F_{S} and F_{T} are the features from student and teacher, respectively. From the formulation, it can be concluded that FKD is simple yet flexible for training. Based on the analysis, we can use the FKD as a criterion to evaluate the searched student network during training.

Teacher Guided Neural Architecture Search

As we claimed before, the pre-defined student network may not be optimal because it requires much effort to design a powerful one. Therefore, we prefer to searching for best candidate. We empirically observe that the student network size is crucial for knowledge distillation. Thus we define our search space as the candidates of width and depth of student network. Suppose X and O are the input and output feature tensors of a convolutional layer (*e.g.*, 3-by-3 convolution), this layer is calculated as the following procedure:

$$O_j = \sum_{k=1}^{c_{in}} X_{k,:,:} * W_{j,k,:,:} \quad 1 \leq j \leq c_{out}, \tag{3}$$

where $W \in R^{c_{out} \times c_{in} \times 3 \times 3}$ indicates the convolutional kernel weight, c_{in} is the input channel, and c_{out} is the output channel. $W_{j,k,:,:}$ corresponds to the k-th input channel and j-th output channel. $*$ denotes the convolutional operation.

Search for width. We use parameters $\alpha \in R^{|C|}$ to indicate the distribution of the possible number of channels in one layer, indicated by C and $\max(C) \leq c_{out}$. The probability of choosing the j-th candidate for the number of channels can be formulated as:

$$p_j = \frac{\exp(\alpha_j)}{\sum_{k=1}^{|C|} \exp(\alpha_k)}, \quad 1 \leq j \leq |C|. \tag{4}$$

However, the sampling operation in the above procedure is non-differentiable (Dong and Yang 2019). Fortunately, the Gumbel-Softmax (Jang, Gu, and Poole 2016; Maddison, Mnih, and Teh 2016; Dong and Yang 2019) to soften the sampling procedure can be used for optimizing α as follows:

$$\hat{p}_j = \frac{\exp((\log(p_j) + o_j)/\tau)}{\sum_{k=1}^{|C|} \exp((\log(p_k) + o_k)/\tau)}, \tag{5}$$

where $o_j = -\log(-\log(\mu))\&\mu \sim \mathcal{U}(0,1)$, and $\mathcal{U}(0,1)$ means the uniform distribution between 0 and 1. τ is the softmax temperature. When $\tau \to 0$, $\hat{p} = [\hat{p}_1, \ldots, \hat{p}_j, \ldots]$ becomes one-shot, and the Gumbel-softmax distribution drawn from $\hat{p}$ becomes identical to the categorical distribution. When $\tau \to \infty$, the Gumbel-softmax distribution becomes a uniform distribution over C. The feature map in our method is defined as the weighted sum of the original feature map fragments with different sizes, where weights are $\hat{p}$. Feature maps with different sizes are aligned by channel wise interpolation (CWI) (Dong and Yang 2019) so as for the operation of weighted sum. To reduce the memory costs, we select a small subset with indexes $I \subseteq [|C|]$ for aggregation instead of using all candidates. Additionally, the weights are re-normalized based on the probability of the selected sizes, which is formulated as $\hat{O}_j =:$

$$\sum_{j \in I} \frac{\exp((\log(p_j) + o_j)/\tau)}{\sum_{j \in I} \exp((\log(p_k) + o_k)/\tau)} \times (O_{1:C_j,:,:}, \max(C_I)) \tag{6}$$

where $I \sim \mathcal{T}_{\hat{p}}$ and $\mathcal{T}_{\hat{p}}$ indicates the multinomial probability distribution parameterized by $\hat{p}$. The involved CWI is a general operation to align feature maps with different sizes, which is implemented via 3D adaptive average pooling operation (He et al. 2015; Dong and Yang 2019). We use Batch Normalization (Ioffe and Szegedy 2015) before CWI to normalize different fragments.

Search for depth. We use parameters $\beta \in R^L$ to indicate the distribution of the possible number of layers in a student network with L convolutional layers. We utilize a similar strategy to sample the number of layers following Eq. (5) and allow β to be differentiable as that of α, using the sampling distribution $\hat{q}_l$ for the depth l. We then calculate the final output feature of the student network as an aggregation from all possible depths, which can be formulated as:

$$O_{out} = \sum_{l=1}^{L} \hat{q}_l \times (\hat{O}_l, C_{out}), \tag{7}$$

where $\hat{O}_l$ indicates the output feature map via Eq. (6) at the l-th layer. C_{out} indicates the maximum sampled channel among all $\hat{O}_l$. The final output feature map O_{out} is fed into a fully connected layer to generate the final face features for evaluation. Obviously, we can back-propagate gradients to both width parameters α and depth parameters β to search different shapes of student network.

Search objective. The final student architecture $\mathcal{A}$ is derived by selecting the candidate with the maximum probability, learned by the architecture parameters A, consisting of α for each layers and β. The goal of our teacher guided neural architecture search (TNAS) is to find an architecture $\mathcal{A}$ with the minimum validation loss $\mathcal{L}_{val}$ after trained by minimizing the training loss $\mathcal{L}_{train}$ as:

$$\min_{\mathcal{A}} \ \mathcal{L}_{val}(w^*_{\mathcal{A}}, \mathcal{A}) \quad w^*_{\mathcal{A}} =_w \ \mathcal{L}_{train}(w, \mathcal{A}), \tag{8}$$

where $w^*_{\mathcal{A}}$ indicates the optimized weights of $\mathcal{A}$. The training loss $\mathcal{L}_{train}$ is the L2 loss of the student network (*i.e.*, Eq. (2)) on training set:

$$\mathcal{L}_{train} = ||F^{train}_{\mathrm{S}} - F^{train}_{\mathrm{T}}||. \tag{9}$$

The validation loss in our search procedure includes not only the L2 loss on validation set but also the penalty for the computation cost:

$$\mathcal{L}_{val} = ||F^{val}_{\mathrm{S}} - F^{val}_{\mathrm{T}}|| + \lambda \mathcal{L}_{cost}, \tag{10}$$

Algorithm 1: Teacher Guided Neural Architecture Search (TNAS)

Input: Unlabelled training set D_{train} and validation set D_{val}; Pre-trained teacher model Θ^{T}; Hyper-parameter λ; Training epochs E.
Initialization: $e = 1$; Initialized weights w_0; Initialized architecture parameters A_0;
while $e \leq E$ **do**
  Sample batch data D_t from D_{train} ;
  Calculate $\mathcal{L}_{train}$ on D_t via Eq. (9) to update student network weights w;
  Sample batch data D_v from D_{val};
  Calculate $\mathcal{L}_{val}$ on D_v via Eq. (10) to update A.
end
Output: Derive the searched student network from A.

	Datasets	#Identities	Images
Training	CASIA-WebFace-R	9,809	0.39M
	MS-Celeb-1M-v1c-R	72,690	3.28M
Test	LFW	5,749	13,233
	SLLFW	5,749	13,233
	CALFW	5,749	12,174
	CPLFW	5,749	11,652
	AgeDB	568	16,488
	CFP	500	7,000
	RFW	11,430	40,607
	MegaFace	530 (P)	1M (G)
	Trillion-Pairs	5,749 (P)	1.58M (G)

Table 1: Face datasets for training and test. (P) and (G) refer to the probe and gallery set, respectively.

where λ is the weight of $\mathcal{L}_{cost}$. The cost loss encourages the computation cost of the student network (*e.g.*, FLOPs) to converge to a target R so that the cost can be dynamically adjusted by setting different R. We used a piece-wise computation cost loss as:

$$\mathcal{L}_{cost} = \begin{cases} \log(E_{cost}(A)) & F_{cost}(A) \geq (1+t)R \\ 0 & (1-t)R < F_{cost}(A) < (1+t)R \\ 1/\log(E_{cost}(A)) & F_{cost}(A) \leq (1-t)R, \end{cases} \tag{11}$$

where $E_{cost}(A)$ computes the expectation of the computation cost, based on the architecture parameters A. Specifically, it is the weighted sum of computation costs for all candidate networks, where the weight is the sampling probability. $F_{cost}(A)$ indicates the actual cost of the searched architecture, whose width and depth are derived from A. $t \in [0, 1]$ denotes a toleration ratio, which slows down the speed of changing the searched architecture. Note that we use FLOPs to evaluate the computation cost of a network, and it is readily to replace FLOPs with other metric, such as latency (Tan et al. 2019; Cai et al. 2018).

Optimization. For clarity, the whole scheme of our teacher guided neural search (TNAS) framework is summarized in Algorithm 1. During searching, we alternatively minimize L_{train} on the training set to optimize the student networks' weights w and L_{val} on the validation set to optimize the architecture parameters A. After searching, we pick up the number of channels with the maximum probability as width and the number of layers with the maximum probability as depth. The final searched student network is constructed by the selected width and depth, whose parameters are then learned by knowledge distillation.

Experiments

Datasets

Training Data. This paper involves two popular training datasets, including CASIA-WebFace (Yi et al. 2014) and MS-Celeb-1M (Guo et al. 2016). Unfortunately, the original CASIA-WebFace and MS-Celeb-1M datasets consist of a great many face images with noisy labels. To be fair, we use the clean version of CASIA-WebFace (Zhao et al. 2019, 2018a) and MS-Celeb-1M-v1c for training.

Test Data. We use nine face recognition benchmarks, including LFW (Huang, Ramesh, and Miller. 2007), SLLFW (Deng et al. 2017), CALFW (Zheng et al. 2017), CPLFW (Zheng et al. 2018), AgeDB (Moschoglou et al. 2017), CFP (Sengupta et al. 2016), RFW (Wang et al. 2018c), MegaFace (Nech and Kemelmacher-Shlizerman 2017) and Trillion-Pairs, as the test data.

Dataset Overlap Removal. In face recognition, it is very important to perform open-set evaluation (Liu et al. 2017; Deng et al. 2019; Wang et al. 2019c), *i.e.*, there should be no overlapping identities between training set and test set. To this end, we need to carefully remove the overlapped identities between the employed training datasets and the test datasets. For the overlap identities removal tool, we use the publicly available script provided by (Wang et al. 2018a) to check whether if two names (one of which is from training set and the other comes from test set) are of the same person. In consequence, we remove 766 identities from the training set CASIA-WebFace and 14,718 identities from MS-Celeb-1M-v1c. For clarity, we denote the refined training datasets as CASIA-WebFace-R and MS-Celeb-1M-v1c-R, respectively. Important statistics of all the involved datasets are summarized in Table 1. To be rigorous, all the experiments are based on the refined training sets.

Experimental Settings

Data Processing. We detect the faces by adopting the FaceBoxes detector (Zhang et al. 2017, 2019) and localize five landmarks (two eyes, nose tip and two mouth corners) through a simple 6-layer CNN (Feng et al. 2018; Liu et al. 2019). The detected faces are cropped and resized to 144×144, and each pixel (ranged between [0,255]) in RGB images is normalized by subtracting 127.5 and divided by 128. For all the training faces, they are horizontally flipped with probability 0.5 for data augmentation.

Pre-trained Teacher Network. There are many kinds of network architectures (Wang et al. 2017; Deng et al. 2019) and several loss functions (Wang et al. 2018a,b; Deng et al. 2019) for face recognition. Without loss of generality, we

Data	Arch.	Knowl.	LFW	MF-Id.	MF-Veri.
	SER100	Teacher	99.73	97.70	98.19
Celeb-R	R50	AM	99.60	89.65	90.85
		PKD	**99.76**	91.85	93.19
		FKD	99.68	**95.52**	**96.43**
CASIA-R	R50	AM	98.24	58.60	63.92
		FKD	**99.63**	**90.75**	**90.96**

Table 2: Performance (%) of different knowledge on the test sets LFW and MegaFace. "Teacher" is pre-trained and frozen. "AM" is directly trained by AM-Softmax loss (Wang et al. 2018a). "PKD" is trained by probability knowledge distillation (*i.e.*,Eq. (1)). "FKD" is trained by feature knowledge distillation (*i.e.*,Eq. (2)).

use SEResNet100 (Deng et al. 2019) as the teacher network. The script is also publicly available at the website[2]. The output gets a 512-dimension feature. We use the MS-Celeb-1M-v1c-R dataset and the AM-Softmax loss (Wang et al. 2018a) for training. For all the experiments in this paper, the teacher network is frozen for searching student network architectures. Here we provide its training set and loss to the competitors KD (Hinton, Vinyals, and Dean. 2015) and FitNet (Romero, Ballas, and Kahou. 2014) for evaluation.

Searching. For searching the expected student networks, we sample the number of channels over $\{0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ of the original number in the pre-defined student network. We search the depth within each convolutional stage. We sample $|I| = 2$ candidates in Eq. (7) during searching. We set the hyper-parameter λ of 2. We optimize the weights via SGD and the architecture parameters via Adam. For the weights, we start the learning rate from 0.1 and reduce it by the cosine scheduler (Loshchilov and Hutter 2016). For the architecture parameters, we use the constant learning rate of 0.001 and a weight decay of 0.001. The toleration ratio t is always set as 5%. The softmax temperature τ in Eq. (5) is linearly decayed from 10 to 0.1. We adopt the CASIA-WebFace-R as the training set and the LFW as the validation set to search student networks.

Training. For training the searched student networks, all of them are trained from scratch by SGD algorithm, with the batch size 256. The weight decay is set to 0.0005 and the momentum is 0.9. The learning rate is initially 0.1. On CASIA-WebFace-R dataset, we empirically divide the learning rate by 10 at 9, 18, 26 epochs and finish the training process at 30 epochs. On MS-Celeb-1M-v1c-R dataset, we divide the learning rate by 10 at 4, 8, 10 epochs, and finish the training process at 12 epochs. For all the compared methods, we run their source codes and keep the same experimental settings. All experiments in this paper are implemented by Pytorch library (Paszke et al. 2019).

Test. At test stage, only the original image features are employed to compose the face representation. All the reported results in this paper are evaluated by a single model, without model ensemble or other fusion strategies. For evaluation metrics, the cosine similarity is utilized. We follow the unrestricted with labelled outside data protocol (Huang, Ramesh, and Miller. 2007) to report the performance on LFW, SLLFW, CALFW, CPLFW, AgeDB, CFP and RFW. On Megaface and Trillion-Pairs, face identification and verification are conducted by ranking and thresholding the scores. Specifically, for face identification, the Cumulative Match Characteristics (CMC) curves are adopted to evaluate the Rank-1 accuracy. For face verification, the Receiver Operating Characteristic (ROC) curves are adopted. The true positive rate (TPR) at low false acceptance rate (FAR) is emphasized since in real applications false acceptance gives higher risks than false rejection.

[2]https://github.com/wujiyang/Face_Pytorch

Data	Arch.	FLOPs	LFW	MF-Id.	MF-Veri.
	SER100	19.96G	99.73	97.70	98.19
Celeb-R	(1/2)R50	0.56G	99.45	87.69	89.71
	TNAS-W	0.56G	**99.63**	**89.10**	**90.95**
	(1)R50	2.22G	99.68	95.52	96.43
	TNAS-W	2.12G	**99.73**	**95.65**	**96.43**
CASIA-R	(1/2)R50	0.56G	98.98	79.02	80.73
	TNAS-W	0.56G	**99.21**	**82.35**	**85.21**
	(1)R50	2.22G	**99.63**	90.75	90.96
	TNAS-W	2.12G	99.58	**90.77**	**91.30**

Table 3: Performance (%) of different widths of ResNet on the test sets LFW and MegaFace. "TNAS-W" means searching for width of our method.

Data	Arch.	FLOPs	LFW	MF-Id.	MF-Veri.
	SER100	19.96G	99.73	97.70	98.19
Celeb-R	R50	2.22G	99.68	95.52	96.43
	TNAS-D	2.21G	**99.71**	**95.86**	**96.45**
CASIA-R	ResNet50	2.22G	**99.63**	90.75	90.96
	TNAS-D	2.21G	99.58	**91.67**	**93.29**

Table 4: Performance (%) of different depths of ResNet on the test sets LFW, SLLFW and MegaFace. "TNAS-D" means searching for depth of our method.

Ablation Study

Effect of different knowledge. We first investigate the effect of different knowledge for face recognition distillation. The performance comparison is reported in Table 2. On the training set MS-Celeb-1M-v1c-R, the feature knowledge distillation FKD achieves higher performance than directly training from scratch (AM) and the probability knowledge distillation PKD in most of cases. On the training set CASIA-WebFace-R, since the training classes of student (9,809) are different from teacher's (72,690), PKD can not be applicable to this case and the performance of AM and FKD is reported. From the values, we can observe that the performance exhibits the same trends, *i.e.*, FKD keeps better than AM. To sum up, the FKD is flexible and powerful than other knowledge for face recognition. Consequently, we can

Method	FLOPs	LFW	SLLFW	CALFW	CPLFW	AgeDB	CFP	RFW			
								Caucasian	Indian	Asian	African
Teacher	19.96G	99.73	99.46	95.61	90.10	98.16	96.52	99.16	97.33	95.16	95.66
AM	0.56G	99.20	97.75	92.26	81.88	94.86	88.77	94.16	85.16	84.16	83.00
FKD	0.56G	99.45	98.36	93.03	80.93	96.15	89.67	93.99	87.33	87.16	84.00
KD	0.56G	99.11	97.40	92.09	80.38	94.25	86.21	93.66	86.66	82.83	82.99
FitNet	0.56G	99.40	97.98	92.26	81.93	95.18	88.70	94.33	87.16	87.66	83.33
S-selection	0.56G	99.58	98.16	93.56	80.96	96.08	89.24	94.33	87.00	84.83	84.16
TNAS-W	0.56G	99.63	98.43	93.71	81.30	96.36	90.12	95.66	87.66	86.66	83.00
TNAS-D	0.59G	99.45	98.45	93.68	81.80	96.16	90.27	95.50	**89.00**	87.83	84.49
TNAS	0.55G	**99.63**	**98.68**	**94.29**	**83.56**	**96.53**	**91.47**	**95.66**	87.83	**88.00**	**85.00**

Table 5: Verification performance (%) of different methods on the test sets LFW, SLLFW, CALFW, CPLFW, SLLFW, AgeDB, CFP and RFW. "TNAS" means searching for both width and depth of our method. The baseline architecture is (1/2)ResNet50 and the training set is MS-Celeb-1M-v1c-R.

use the FKD loss as the search objective (*i.e.*, Eqs. (9) and (10)) for searching the target student networks.

Effect of different architectures. We further evaluate the performance with different architectures. To begin with, we introduce the baseline architecture ResNet, which consists of 4 stages and starts with 16 channels, and [32, 64, 128, 256] in the corresponding 4 stages. For example, (1/2)ResNet50 means that the depth is 50, and the channels is reduce to 0.5 on all convolutional layers, *i.e.*, starts with 8, and changes into [16, 32, 64, 128] in the 4 stages. For more details, the adopted ResNet architectures are provided in supplementary materials. The performance comparison is reported in Tables 3 and 4. From the values, it can be conclude that different architectures may affect the performance heavily. The networks with smaller sizes like (1/2)ResNet50 and ResNet18 exhibit lower capability for face recognition. For searching width, we can see that our searched architectures TNAS-W with similar flops to the pre-defined student networks, outperform the pre-defined ones by a large margin, especially when the expected flops is low. For searching depth, the improvement of our TNAS-D is not very obvious. The reason behind this is that the search space of depth is usually small. To balance flops and performance, we employ (1/2)ResNet50 as baseline in the following experiments unless otherwise specified.

Results on LFW, SLLFW, CALFW, CPLFW, AgeDB and CFP

The left part of Tables 5 and 6 show the results of different approaches on LFW, SLLFW, CALFW, CPLFW, AgeDB and CFP test sets. The bold number in each column represents the best result. From the values, we observe that most of the knowledge distillation methods with pre-defined student network are better than simply training the student network from scratch. Among all the competitors, the FKD and S-selection (Luo et al. 2016) seem to be more flexible and achieve higher performance than others. However, the hand-crafted student network hinders their superiority. For our method TNAS, it beats the best competitor S-selection in most of cases on all these test sets because of the searched student network. From the results, we can also observe that searching for width or depth of student network individually may not achieve the best performance. If we jointly search for both width and depth (*i.e.*, TNAS) of student networks, we can further boost the performance with similar flops.

Results on RFW

The right part of Tables 5 and 6 display the performance comparison of all the methods on the RFW test set. RFW is a face recognition benchmark for measuring racial bias, which consists of four test subsets, namely Caucasian, Indian, Asian and African. The results exhibit the same trends that emerged on previous test sets. Concretely, most of the knowledge distillation methods are consistently better than directly training the student network from scratch (*i.e.*, AM). For instance, FitNet achieves higher performance than AM, especially on the subsets Indian and Asian. S-selection beats the baseline AM in all the test subsets. While for our teacher guided neural architecture search TNAS, it can further boost the performance because the student network is searched from the large design space.

Results on MegaFace and Trillion-Pairs

Tables 7 and 8 give the identification and verification results of different methods on MegaFace and Trillion-Pairs challenge. In particular, compared with directly training the hand-crafted student network with one-hot labels, *i.e.*, AM, most of competitors (*e.g.*, FitNet and S-selection) have shown their strong abilities to achieve better performance. For our teacher guided neural architecture search method TNAS, we can further boost the performance because of the searched student network with expected flops. The improvement is large on these two test sets, especially at the small rank and at the very low false alarm rate. In Figures 1 and 2, we draw the CMC curves to evaluate the performance of face identification and the ROC curves to evaluate the performance of face verification on MegaFace Set 1. From the curves, we can see the similar trends at other measures. On Trillion-Pairs, we can observe that the results exhibit the same trends that emerged on MegaFace test set.

Method	FLOPs	LFW	SLLFW	CALFW	CPLFW	AgeDB	CFP	RFW			
								Caucasian	Indian	Asian	African
Teacher	19.96G	99.73	99.46	95.61	90.10	98.16	96.52	99.16	97.33	95.16	95.66
AM	0.56G	97.98	92.31	84.61	75.93	87.45	88.28	86.00	78.00	76.16	75.83
FKD	0.56G	98.98	96.73	90.85	82.48	93.50	92.77	90.99	83.50	82.33	80.00
S-selection	0.56G	99.06	96.98	90.80	82.48	93.38	92.61	92.16	84.66	82.50	79.66
TNAS-W	0.56G	99.21	97.06	91.01	82.75	94.26	92.81	91.16	**85.50**	81.83	78.83
TNAS-D	0.59G	99.35	97.23	91.62	83.33	94.63	93.11	**93.00**	84.33	81.46	76.00
TNAS	0.55G	**99.36**	**97.28**	**91.63**	**83.63**	**94.66**	**93.77**	92.83	85.33	**83.16**	**81.50**

Table 6: Verification performance (%) of different methods on the test sets LFW, SLLFW, CALFW, CPLFW, SLLFW, AgeDB, CFP and RFW. "TNAS" means searching for both width and depth of our method. The baseline architecture is (1/2)ResNet50 and the training set is CASIA-WebFace-R.

Method	FLOPs	MF-Id.	MF-Veri.	TP-Id.	TP-Veri.
Teacher	19.96G	97.70	98.19	75.01	72.73
AM	0.56G	80.46	82.44	30.46	30.53
FKD	0.56G	87.69	89.71	40.62	39.80
KD	0.56G	77.51	79.45	28.19	28.44
FitNet	0.56G	81.86	83.79	39.81	32.74
S-selection	0.56G	87.62	88.22	39.32	38.88
TNAS-W	0.56G	89.10	90.95	39.42	36.96
TNAS-D	0.59G	89.29	89.94	39.60	37.34
TNAS	0.55G	**90.40**	**92.85**	**42.45**	**40.84**

Table 7: Performance (%) of different methods on the test sets MegaFace and Trillion-Pairs. The training set is MS-Celeb-1M-v1c-R.

Method	FLOPs	MF-Id.	MF-Veri.	TP-Id.	TP-Veri.
Teacher	19.96G	97.70	98.19	75.01	72.73
AM	0.56G	51.74	56.49	2.83	0.44
FKD	0.56G	79.02	80.73	20.24	18.59
S-selection	0.56G	79.21	81.76	20.38	18.52
TNAS-W	0.56G	82.35	85.21	22.17	18.95
TNAS-D	0.59G	**82.55**	**85.99**	22.14	**19.38**
TNAS	0.55G	81.81	85.19	**22.19**	19.12

Table 8: Performance (%) of different methods on the test sets MegaFace and Trillion-Pairs. The training set is CASIA-WebFace-R.

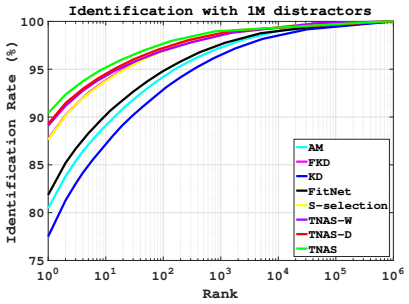

Figure 1: CMC curves of different methods with 1M distractors on MegaFace Set 1. The training set is MS-Celeb-1M-v1c-1M-R.

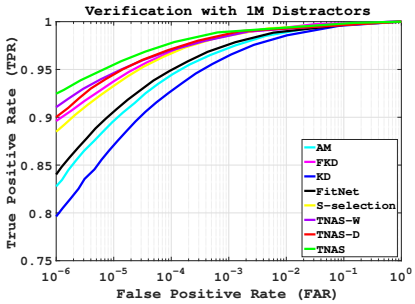

Figure 2: ROC curves of different methods with 1M distractors on MegaFace Set 1. The training set is MS-Celeb-1M-v1c-1M-R.

Conclusion

In this paper, we have proposed a novel teacher guided neural architecture search method for face recognition. Specifically, based on the observation that the feature knowledge is more flexible and powerful for face recognition distillation, we develop a novel search objective, which can enhance the capability and the performance of student network very well. Moreover, we define the search space as the candidates of student network with flexible channel and layer sizes. Extensive experiments on a variety of face recognition benchmarks have validated the effectiveness of our new approach over the state-of-the-art alternatives.

References

Aguinaldo, A.; Chiang, P.-Y.; Gain, A.; Patil, A.; Pearson, K.; and Feizi, S. 2019. Compressing GANs using Knowledge Distillation. *arXiv preprint arXiv:1902.00159* .

Bashivan, P.; Tensen, M.; and DiCarlo, J. J. 2019. Teacher guided architecture search. In *Proceedings of the IEEE International Conference on Computer Vision*, 5320–5329.

Cai, H.; Chen, T.; Zhang, W.; Yu, Y.; and Wang, J. 2018. Efficient architecture search by network transformation. In *Thirty-Second AAAI conference on artificial intelligence*.

Chen, S.; Liu, Y.; Gao, X.; and Han, Z. 2018. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In *CCBR*, 428–438. Springer.

Chen, T.; Goodfellow, I.; and Shlens, J. 2015. Net2net: Accelerating learning via knowledge transfer. *arXiv preprint arXiv:1511.05641* .

Chen, X.; Xie, L.; Wu, J.; and Tian, Q. 2019. Progressive DARTS: Bridging the Optimization Gap for NAS in the Wild. *arXiv preprint arXiv:1912.10952* .

Chen, Y.; Wang, N.; and Zhang, Z. 2018. Darkrank: Accelerating deep metric learning via cross sample similarities transfer. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

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

Deng, W.; Hu, J.; Chen, B.; and Guo, J. 2017. Fine-grained face verification: Fglfw database, baselines, and human-dcmn partnership. *Pattern Recognition* 66: 63–73.

Dong, X.; and Yang, Y. 2019. Network pruning via transformable architecture search. In *Advances in Neural Information Processing Systems*, 759–770.

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

Feng, Z.-H.; Kittler, J.; Awais, M.; Huber, P.; and Wu, X.-J. 2018. Wing loss for robust facial landmark localisation with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2235–2245.

Ge, S.; Zhao, S.; Li, C.; and Li, J. 2018. Low-resolution face recognition in the wild via selective knowledge distillation. *IEEE Transactions on Image Processing* 28(4): 2051–2062.

Gordon, A.; Eban, E.; Nachum, O.; Chen, B.; Wu, H.; Yang, T.-J.; and Choi, E. 2018. Morphnet: Fast & simple resource-constrained structure learning of deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1586–1595.

Guo, Y.; Zhang, L.; Hu, Y.; He, X.; and Gao, J. 2016. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision*, 87–102. Springer.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence* 37(9): 1904–1916.

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

Huang, G.; Ramesh, M.; and Miller., E. 2007. Labeled faces in the wild: A database for studying face recognition in unconstrained enviroments. *Technical Report* .

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* .

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* .

Karlekar, J.; Feng, J.; and Pranata, S. 2019. Deep Face Recognition Model Compression via Knowledge Transfer and Distillation. *arXiv preprint arXiv:1906.00619* .

Kemelmacher-Shlizerman, I.; Seitz, S. M.; Miller, D.; and Brossard, E. 2016. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4873–4882.

Li, J.; Zhao, J.; Zhao, F.; Liu, H.; Li, J.; Shen, S.; Feng, J.; and Sim, T. 2016. Robust face recognition with deep multi-view representation learning. In *Proceedings of the 24th ACM international conference on Multimedia*, 1068–1072.

Li, Q.; Jin, S.; and Yan, J. 2017. Mimicking very efficient network for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6356–6364.

Liang, X.; Wang, X.; Lei, Z.; Liao, S.; and Li, S. Z. 2017. Soft-margin softmax for deep classification. In *International Conference on Neural Information Processing*, 413–421. Springer.

Liu, H.; and Simonyan, K. 2018. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.090533* .

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

Liu, Y.; Jia, X.; Tan, M.; Vemulapalli, R.; Zhu, Y.; Green, B.; and Wang, X. 2020. Search to Distill: Pearls are Everywhere but not the Eyes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7539–7548.

Liu, Y.; Si, Y.; Wang, X.; and Mei, T. 2019. A High-Efficiency Framework for Constructing Large-Scale Face Parsing Benchmark. *arXiv preprint arXiv:1905.04830* .

Loshchilov, I.; and Hutter, F. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* .

Luo, P.; Zhu, Z.; Liu, Z.; Wang, X.; and Tang, X. 2016. Face model compression by distilling knowledge from neurons. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Maddison, C. J.; Mnih, A.; and Teh, Y. W. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712* .

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

Nech, A.; and Kemelmacher-Shlizerman, I. 2017. Level playing field for million scale face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7044–7053.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703* .

Peng, B.; Jin, X.; Liu, J.; Li, D.; Wu, Y.; Liu, Y.; Zhou, S.; and Zhang, Z. 2019. Correlation Congruence for Knowledge Distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, 5007–5016.

Romero, A.; Ballas, N.; and Kahou., S. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550* .

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

Sengupta, S.; Chen, J.-C.; Castillo, C.; Patel, V. M.; Chellappa, R.; and Jacobs, D. W. 2016. Frontal to profile face verification in the wild. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1–9. IEEE.

Sun, Y.; Chen, Y.; Wang, X.; and Tang, X. 2014. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, 1988–1996.

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

Wang, C.; Lan, X.; and Zhang, Y. 2017. Model Distillation with Knowledge Transfer from Face Classification to Alignment and Verification. *arXiv preprint arXiv:1709.02929* .

Wang, F.; Cheng, J.; Liu, W.; and Liu, H. 2018a. Additive margin softmax for face verification. *IEEE Signal Processing Letters* 25(7): 926–930.

Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; and Tang, X. 2017. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3156–3164.

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

Wang, M.; Deng, W.; Peng, J.; Tao, X.; and Huang, Y. 2018c. Racial Faces in-the-Wild: Reducing Racial Bias by Deep Unsupervised Domain Adaptation. *arXiv:1812.00194* .

Wang, M.; Liu, R.; Hajime, N.; Narishige, A.; Uchida, H.; and Matsunami, T. 2019a. Improved Knowledge Distillation for Training Fast Low Resolution Face Recognition Model. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 0–0.

Wang, X.; Fu, T.; Liao, S.; Wang, S.; Lei, Z.; and Mei, T. 2020a. Exclusivity-Consistency Regularized Knowledge Distillation for Face Recognition. In *Proceedings of the European Conference on Computer Vision*, 1–8.

Wang, X.; Wang, S.; Chi, C.; Zhang, S.; and Mei, T. 2020b. Loss function search for face recognition. In *International Conference on Machine Learning*, 10029–10038. PMLR.

Wang, X.; Wang, S.; Wang, J.; Shi, H.; and Mei, T. 2019b. Co-mining: Deep face recognition with noisy labels. In *Proceedings of the IEEE international conference on computer vision*, 9358–9367.

Wang, X.; Wang, S.; Zhang, S.; Fu, T.; Shi, H.; and Mei, T. 2018d. Support Vector Guided Softmax Loss for Face Recognition. *arXiv:1812.11317* .

Wang, X.; Zhang, S.; Lei, Z.; Liu, S.; Guo, X.; and Li, S. Z. 2018e. Ensemble Soft-Margin Softmax Loss for Image Classification. *arXiv preprint arXiv:1805.03922* .

Wang, X.; Zhang, S.; Wang, S.; Fu, T.; Shi, H.; and Mei, T. 2019c. Mis-classified Vector Guided Softmax Loss for Face Recognition. *arXiv preprint arXiv:1912.00833* .

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

Yi, D.; Lei, Z.; Liao, S.; and Li., S. Z. 2014. Learning face representation from scratch. *arXiv:1411.7923.* .

Zhang, S.; Wang, X.; Lei, Z.; and Li, S. Z. 2019. Faceboxes: A CPU real-time and accurate unconstrained face detector. *Neurocomputing* .

Zhang, S.; Zhu, X.; Lei, Z.; Shi, H.; Wang, X.; and Li, S. Z. 2017. Faceboxes: A CPU real-time face detector with high accuracy. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, 1–9. IEEE.

Zhao, J.; Cheng, Y.; Xu, Y.; Xiong, L.; Li, J.; Zhao, F.; Jayashree, K.; Pranata, S.; Shen, S.; Xing, J.; et al. 2018a. Towards pose invariant face recognition in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2207–2216.

Zhao, J.; Li, J.; Tu, X.; Zhao, F.; Xin, Y.; Xing, J.; Liu, H.; Yan, S.; and Feng, J. 2019. Multi-prototype networks for unconstrained set-based face recognition. *arXiv preprint arXiv:1902.04755* .

Zhao, J.; Xiong, L.; Karlekar Jayashree, P.; Li, J.; Zhao, F.; Wang, Z.; Sugiri Pranata, P.; Shengmei Shen, P.; Yan, S.; and Feng, J. 2017. Dual-agent gans for photorealistic and identity preserving profile face synthesis. *Advances in neural information processing systems* 30: 66–76.

Zhao, J.; Xiong, L.; Li, J.; Xing, J.; Yan, S.; and Feng, J. 2018b. 3d-aided dual-agent gans for unconstrained face recognition. *IEEE transactions on pattern analysis and machine intelligence* 41(10): 2380–2394.

Zheng, T.; Deng, W.; Hu, J.; and Hu, J. 2017. Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv:1708.08197* .

Zheng, T.; Deng, W.; Zheng, T.; and Deng, W. 2018. Crosspose lfw: A database for studying crosspose face recognition in unconstrained environments. *Tech. Rep* .