This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3106483, IEEE Access

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Learning to Drop Expensive Layers for Fast Face Recognition

JUNHUI LI[1], WEI JIA[2], YAN HU[1], SHOUQING LI [3], AND XIAOGUANG TU[1]
[1]Civil Aviation Flight University of China, Aviation Engineering Institute, 46 Nanchang Road, Guanghan, China, 618307
[2]Huazhong University of Science and Technology, School of Cyber Science and Engineering, Luoyu Road 1037, Wuhan, China, 430074.
[3]Civil Aviation Flight University of China, Key Laboratory of Flight Techniques and Flight Safety, CAAC, 46 Nanchang Road, Guanghan, China, 618307

Corresponding author: Xiaoguang Tu (e-mail: xguangtu@outlook.com).

This work was partially supported by the Open Fund Project of Key Laboratory of Flight Technology and Flight Safety of CAFUC (FZ2020KF10), the National Science Foundation of China (62006244), the Project of Comprehensive Reform of Electronic Information Engineering Specialty for Civil Aircraft Maintenance (14002600100017J172), the Project of Civil Aviation Flight University of China (Grant Nos. J2018-56, CJ2019-03, J2020-060), and the Sichuan University Failure Mechanics & Engineering Disaster Prevention and Mitigation Key Laboratory of Sichuan Province Open Foundation (2020FMSCU02).

ABSTRACT Recent years have seen many advances based on Deep Convolutional Neural Networks (DCNNs) in the tasks of face recognition, most of which are developed to pursue high recognition accuracy. In this paper, we propose a novel Fast FAce Recognizer (Fast-FAR), learning to improve the speed of DCNN-based face recognition model without sacrificing recognition accuracy. Our fundamental insight is that the computation increases exponentially with the depth of a network, the easily identifiable face images can be accurately recognized by the cheep features (pixel values at shallow layers), while the challenging samples that exhibit low quality, large pose variations or occlusions need to be processed by the expensive deep layers. The major contribution of this paper is the Reinforcement Learning Agent (RLA), which is proposed to learn a decision policy determined by a reward function. The policy adaptively decides whether the recognition should be performed at an early layer with a high recognition confidence, or proceeding to the subsequent layers, thus significantly reducing feed-forward cost for the easy faces. According to the extensive experiments on the popular face recognition benchmarks, Fast-FAR reduces the inference time by 14.22%, 20.61%, and 7.84% on the dataset LFW, AgeDB-30 and CFP-FP, respectively.

INDEX TERMS Fast Face Recognition, Reinforcement Learning, Deep Convolutional Neural Networks

I. INTRODUCTION

Face recognition has made great progress in recent years, owing to the advancement of Deep Convolutional Neural Networks (DCNNs). With the works DeepID [1] and DeepFace [2] firstly used to automatically learn features on the large scale face datasets, DCNN-based methods have dominated the field of face recognition. Some of the works like DeepID2+ [3] and DeepID3 [4] focus on developing advanced network structures to boost face recognition performance. Recent works [5, 6, 7, 8, 9, 10, 11, 12, 22] mainly explore the design of loss functions to enhance the representation ability for the learned features. FaceNet [13] uses the triplet loss to supervise the embedding learning, obtaining the state-of-the-art face recognition performance. Later, Wen *et al.* [6] propose a center loss to compact the intra-class clusters to the center of each identity. L-Softmax [5] adds angular constraint to each identity to learn discriminative features. SphereFace [8] assumes that the linear transformation matrix in the last fully-connected layer can be used as a representation of the class centres in an angular space, and proposes the Angular Softmax (A-Softmax) loss to impose discriminative constraint on a hypersphere manifold. CosFace [9] reformulates the softmax as a cosine loss, and introduces a cosine margin to further maximize the decision margin in the angular space. In the very recent work [10], Deng *et al.* have proposed the Additive Angular Margin Loss (ArcFace). They calculate the angle between the feature and the target weight (center for each class), and then add an angular margin penalty to the target angel on the angular space. ArcFace achieves the best state-of-the-art face recognition performance to date with more stable training of the network.

It seems most of the previous works are devoted to the improvement of face recognition accuracy, only few of them are proposed to reduce the recognition time. In the work [25], Guo *et al.* propose a meta learning approach for face

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3106483, IEEE Access

recognition by building the domain-shift batches through a domain-level sampling strategy and apply back-propagated gradients/metagradients on synthesized source/target domains by optimizing multi-domain distributions. Later, Chang *et al.* [24] apply data uncertainty learning to face recognition, performing feature (mean) and uncertainty (variance) learning simultaneously. Deng *et al.* propose an improved version for Arcface [10], which encourages one dominant sub-class that contains the majority of clean faces and non-dominant sub-classes that include hard or noisy faces. In the work [22], Tu *et al.* develop a Multi-Degradation Face Restoration model which can address face frontalization and restoration simultaneously for face recognition.

To improve the recognition efficiency, Wu *et al.* [14] argue that the labels for current training face images from the internet are ambiguous and inaccurate, and propose a Light CNN to learn a compact embedding on the large-scale training data with the noisy labels, towards faster and more accurate face recognition. Specifically, they introduce a special case of maxout, *i.e,* the Max-Feature-Map (MFM) operation, into each convolutional layer of a DCNN. The MFM works as a separator to purify the informative signals from the noisy data, as well as a filter to perform feature selection. Experimental results have show that the light CNN can utilize large-scale noisy data to learn a Light model that is efficient in computational resources and storage spaces. However in the work [15], De *et al.* propose to accelerate face recognition by the distillation technology, which transfers the similarity information of a teacher network to a small model (student network) by adaptively varying the margin between positive and negative pairs. According to their reported results, the method achieves a faster processing rate (>10) and a lower memory occupation (1/6) on the *dlib-resnet-v1* face recognition model. However, the obtained face recognition performance drops to some extent compared with the complex teacher model.

Due to the high demand on real-time recognition, and the computation limitation of many mobile devices such as laptop and cell phones, the efficiency of DCNN-based face recognition approaches still needs to be improved. In this paper, we propose a generic framework, *i.e.,* Fast FAce Recognizer (Fast-FAR), aiming to reduce the recognition time for an arbitrary DCNN-based face recognition model. Typically, the recognition difficulty varies across face images, face images with small pose variations and good visual quality can be easily recognized by early layers of a network. A deeper layer contains more parameters compared with a shallow layer, therefore it occupies more computational resources. If the subsequent layers can be saved for the easy face images, the recognition time can be significantly reduced. Based on this observation, we propose to adaptively learn a decision for the recognition layer via reinforcement learning. Specifically, our Face-FAR contains a main network to learn discriminative representations for face images, and two sub-networks, *i.e.,* the Embedding sub-Network (E-Net) to compress the feature of different dimensions to a vector with fixed length in the unified feature representing space, the Decision sub-Network (D-Net) to determine whether the recognition should be performed at current layer or proceed to the next layer. The Reinforcement Learning Agent (RLA) is used to examine the state of each layer at each step and decide on the action (stop or proceed) by a reward function.

We apply our fast-FAR model to the wildly used CNN backbone ResNet-50 to perform face recognition on various face recognition benchmarks. Extensive experiments have shown that fast-FAR saves computational burdens at least 7.8% for all the benchmarks during inference, **while still achieving state-of-the-art face recognition performance.**

II. FAST FACE RECOGNITION

In this section, we explain our method in details. We first give an overview for the proposed model and then describe reinforcement learning on deep layer selection.

A. MODEL OVERVIEW

Our Fast-FAR contains a main network and two sub-networks, *i.e.,* the Embedding sub-Network (E-Net) and the Decision sub-Network (D-Net). The main network ResNet-50 (B) is used to learn discriminative features for face recognition. E-Net E is used to convert an arbitrary feature from each layer of B into an embedding space with fixed-length, therefore the converted features are comparable in the embedding space. D-Net produces two actions (stop or proceed) from the converted features by maximizing the sum of expected rewards on a given face image, to decide whether the input face can be accurately recognized on the early layer of the network. Fig. 1 illustrates the architecture of Fast-FAR.

The main network contains 4 blocks (B_1, ..., B_4) to generate high-level discriminative features, the dimensions of the outputs from the 4 blocks are 56 $\times$ 56, 28 $\times$ 28, 14 $\times$ 14, and 7 $\times$ 7, respectively. In the next step, the outputs of the 4 blocks will be taken as inputs by the D-Net, to compare with each other, determinating which one is better for recognition. However, the dimensions of the outputs from different layers of the main network are different. To make them comparable, we design the E-Nets (E_1, ..., E_3), which are connected to the first three blocks of the main network (B_1, ..., B_3), to convert the output features from different layers into the same embedding space with a fixed size 7 $\times$ 7, *i.e.,* the feature space of B_4. Actually the dimension of the features from different blocks are predefined, the dimension of the output features has no direct relationship with the number of layers. In this work, we use ResNet-50 as the main backbone for feature learning. However, we can use other popular networks as the backbones or dividing the main backbone into different sub-networks, then the dimension of the output features can be different. The architecture of E-Net is illustrated in Table 1.

Hence, we propose the embedding loss $\mathcal{L}_\mathrm{e}$ to draw the converted features closer to the feature of the last convolu-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3106483, IEEE Access

TABLE 1: The architecture of E-Net (E_1, E_2, E_3). E_1 has 4 convolutional layers, E_2 has 3 convolutional layers, while E_3 has 1 convolutional layer. [ks, fm, s] represents kernel size, feature map number and stride, respectively.

	Layer 1 / [ks, fm, s]	Layer 2 / [ks, fm, s]	Layer 3 / [ks, fm, s]	Layer 4 / [ks, fm, s]
E_1	[3x3,256,s=2] [3x3,256,s=1]	[3x3,256,s=2] [3x3,256,s=1]	[3x3,256,s=1] [3x3,256,s=1]	[3x3,512,s=2] [3x3,512,s=1]
E_2	-	[3x3,256,s=2] [3x3,256,s=1]	[3x3,256,s=1] [3x3,256,s=1]	[3x3,256,s=2] [3x3,256,s=1]
E_3	-	-	-	[3x3,256,s=2] [3x3,256,s=1]

tional layer. For a main network that has M convolutional blocks, $\mathcal{L}_{\text{e}}$ is defined as

$$\mathcal{L}_{\text{e}} = \frac{1}{N}\sum_{i=1}^{M-1}\sum_{j=1}^{N}(E(f_{i,j}) - f_j)^2,$$

where $M-1$ denotes the first $M-1$ blocks of the main network, N denotes the sample number of one mini-batch, $E(\cdot)$ represents feature converting by the E-Net, $f_{i,j}$ is the feature produced by the j-th sample in the i-th block, and f_j is the feature of j-th sample produced by the last layer.

The loss $\mathcal{L}_{\text{e}}$ ensures E-Net produce features similar with that of the last block. However, as no identity information is imposed on the converted features, they can hardly discriminate face identities. To this end, we introduce the discrimination loss $\mathcal{L}_{\text{d}}$ to enhance the discrimination ability for the converted features in the embedding space. $\mathcal{L}_{\text{d}}$ is defined as

$$\mathcal{L}_{\text{d}} = \sum_{i=1}^{M}\mathcal{L}_{\text{Arc}}(f_i),$$

where f_i is the converted feature from i-th block of the main network, and $\mathcal{L}_{\text{arc}}$ denotes the ArcFace [10] loss function. Different from traditional softmax loss, ArcFace loss normalizes the bias to 0 and the length of weights and embedding features to 1 by l_2 norm, simplifying the original linear mapping of softmax loss to $s\cos(\theta_j)$ which is expressed as

$$\mathcal{L}_{\text{Arc}} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{s(cos(\theta_{y_i}+m))}}{e^{s(cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n} e^{scos\theta_j}},$$

where m is the angle margin, N and n are the batch size and the class number, respectively, and s is the estimate re-scale value of embedding features before and after normalization. ArcFace enhances the intra-class compactness and inter-class discrepancy by adding an additive angular margin penalty m on the target (ground truth) angle, which can significantly improve the discriminative power for the learned features for face recognition. By employing $\mathcal{L}_{\text{arc}}$ on the embedding space, we obtain $\mathcal{L}_{\text{e}}$, making all the converted features dropped into the same identity metric space with small intra-class distance and large inter-class distance. Therefore, the overall loss function for the converted features is:

$$\mathcal{L}_{\text{c}} = \mathcal{L}_{\text{e}} + \lambda\mathcal{L}_{\text{d}}$$

where λ is the weight constants of the two loss functions. When the combination loss $\mathcal{L}_{\text{c}}$ is smaller than 0.001, the training can be stopped.

B. LEARNING TO DROP EXPENSIVE LAYERS

The D-Nets (D_1, ..., D_3) takes as input the fixed dimension features that converted by E-Nets (E_1, ..., E_3), and decides whether the learning should stop at current layer or proceed to the next layer. During training, the feature extraction at each block has two options, *i.e.,* stop and use the current feature for face recognition, or proceed to the next block for feature extraction. It can be viewed as a Markov Decision Process (MDP), where an agent can make two actions (stop or continue). The final goal is to find an earliest layer that can accurately recognize the input face image. We propose to train a our Fast-FAR end to end by the Q-learning algorithm of deep Reinforcement Learning (RL), which contains a set of states S and actions A, and a reward function R. At each step at the l-th block, the agent checks the current state S_l and takes an action from A_l, to decide whether performing face recognition using the current block, or proceeding to the next block. The reward function R makes the agent learn the best decision to select action and balance the recognition accuracy (using deeper layers) and speed (stop earlier if effective enough).

In our model, the state S_l is the feature map F_l at l-th block. The action set A includes one stop action and one continue action. The reward R function is defined as

$$R(S_l, S_{l+1}) = \begin{cases} 1 & \{k| \max\limits_{k=1,\ldots,N} W_k^T f_l + b_k\} = g \quad \& \quad A = \text{stop} \\ -1 & \{k| \max\limits_{k=1,\ldots,N} W_k^T f_l + b_k\} = g \quad \& \quad A = \text{continue} \\ 1 & \{k| \max\limits_{k=1,\ldots,N} W_k^T f_l + b_k\} \neq g \quad \& \quad A = \text{continue} \\ -1 & \{k| \max\limits_{k=1,\ldots,N} W_k^T f_l + b_k\} \neq g \quad \& \quad A = \text{stop} \end{cases}$$

For the k-th face image from the class g in one mini-batch, f_l denotes the corresponding converted feature in the embedding space, W_k and b_k are the weight and bias in the probability layer, respectively. $\{k| \max\limits_{k=1,\ldots,N} W_k^T f + b_k\}$ is the maximal conditional probability, and N denotes the number of classes.

Q-learning algorithm learns an estimated value that ap-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3106483, IEEE Access

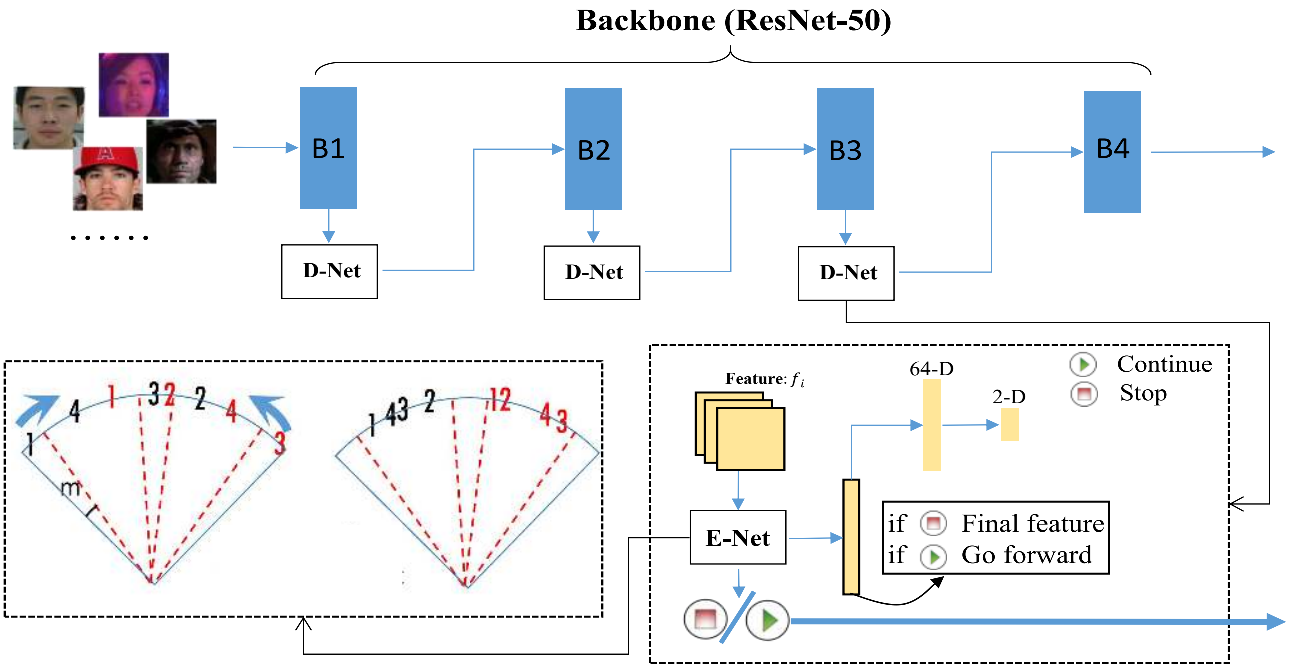

FIGURE 1: Overview of the proposed method. The backbone is divided into 4 blocks, *i.e.*, B_1-B_4. The three Decision sub-Networks D_1-D_3 are connected to the corresponding main blocks B_1-B_3, while E_1, E_2, and E_3 are embedded into the D_1, D_2, and D_3 respectively for feature conversion. The backbone takes images as input and generate feature maps at each block (B_1-B_4). The E-Net converts an arbitrary feature from each layer of B_i into an embedding space with fixed-length for comparison. The D-Net makes a decision whether the learning should stop at current layer or proceed to the next layer.

TABLE 2: Ablation study results by using different loss combinations. "M" represents the main network (ResNet-50), "B_1 - B_4" represent the 4 blocks of "M", respectively. The number in each column of "B_1 - B_4" represents the images that processed by each block of "M". "Acc" means the face recognition accuracy, and "Time" represents the recognition time.

Datasets	Loss combinations	Acc (%)	Time (ms)	B_1	B_2	B_3	B_4
LFW [17]	M	99.55	16.25	0	0	0	12000
	M + E-Net + $\mathcal{L}_e$	99.50	15.43	15	493	5159	6333
	M + E-Net + $\mathcal{L}_d$	99.50	14.80	36	2232	4104	5628
	M + E-Net + $\mathcal{L}_e + \mathcal{L}_d$	99.58	13.94	1006	2540	5398	3056
AgeDB-30 [19]	M	97.33	16.93	0	0	0	12000
	M + E-Net + $\mathcal{L}_e$	97.55	16.57	8	537	5485	5970
	M + E-Net + $\mathcal{L}_d$	96.07	14.87	222	2475	3571	5732
	M + E-Net + $\mathcal{L}_e + \mathcal{L}_d$	97.03	13.44	1325	2469	5417	2789
CFP-FP [20]	M	87.86	18.50	0	0	0	14000
	M + E-Net + $\mathcal{L}_e$	87.50	18.12	23	368	3933	9676
	M + E-Net + $\mathcal{L}_d$	84.57	17.59	40	1954	2686	9320
	M + E-Net + $\mathcal{L}_e + \mathcal{L}_d$	87.60	17.05	869	2121	3021	7989

proaches the real one. In our model, the estimated value is the max probability value of a set of actions ($\max_{s=0,1} a^s$), and the real value is the rewards. The learning process iteratively updates the action-selection policy by:

$$Q(S_l, A_l) = R_l + \gamma \max_{A'} Q(S', A'),$$

where $Q(S_l, A_l)$ means the estimated state when taking action A_l at state S_l, R_l is the overall rewards from the initial state, $\max_{A'} Q(S', A')$ denotes the maximal action reward from state S_l to S_{l+1} and γ is the discount factor, The state $Q(S, A)$ is learned by D-Net.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3106483, IEEE Access

We train D-Net using the following loss function:

$$\mathcal{L}_{\rm p} = \frac{1}{N}\sum_{i=1}^{3}\sum_{j=1}^{N}(\max_{s=0,1} Q^s_{i,j} - R(S_l, S_{l+1}))^2,$$

where $Q^s_{i,j}$ denotes the estimated state taking k-th action for the j-th sample at i-th block.

The training process of Q-learning is described by the the pseudo-code in algorithm 1.

Algorithm 1 Training process of Q-learning

Q-learning:
Initialization: Initialize $Q(S_0, A_0)$ by random values between 0 and 1.
while not converge **do**
  Repeat (for each step of episode)
    Choose an action a from s using the policy of Q-learning.
    Take action a (a = 0 or 1), observe Q
    $Q(S_l, A_l) = R_l + \gamma \max_{A'} Q(S', A')$
  Calculate loss function $\mathcal{L}_{\rm p}$
 end if $\mathcal{L}_{\rm p} < \epsilon$, where ϵ is a small value.

III. EXPERIMENTS

A. IMPLEMENTATION DETAILS AND DATASETS

a: Implementation

Throughout the experiments, the size of face images are fixed as 128 × 128; the constraint factor λ and discount factor γ are fixed as 1 and 0.5, respectively; the batch size is set to 8; the initial learning rate lr for the main network, E-Net and D-Net are set to 0.001, 0.0001 and 0.0001, respectively, lr decreases 10 times at every 2 epochs. Our model is implemented by Pytorch, using one GTX 1080ti (12G) GPU. The model is trained iteratively by the following three steps until convergence. 1. Train the main backbone using ArcFace [10] loss; 2. Fix the parameters of the main network and train E-Net. 3. Fix the parameters of the main network and E-Net, train D-Net.

b: Datasets

We train our model on the MS1MV2 dataset, which is semi-automatically refined from the MS-Celeb-1M [16] dataset. The testing dataset includes LFW [17], AgeDB-30 [19], abd CFP-FP [20]. LFW contains 13233 images from 5749 subjects, 6,000 image pairs are randomly selected for face verification. AgeDB-30 contains 16,488 images from 568 subjects. We evaluate on the age-invariant face verification protocols, which has 10 folds each with 300 intra-class and 300 intra-class pairs. CFP-FP consists of 500 subjects, each with 10 frontal and 4 profile images. We evaluate on the frontal vs. profile protocol, which contains 3,500 positive pairs and 3,500 negative pairs.

B. ABLATION STUDY

We first evaluate different loss combinations for E-Net to reveal their effectiveness in our model. We consider four combination variants, the main network without E-Net and D-Net (only the ArcFace loss is used) and three Fast-FAR variants, *i.e.,* the main network with E-Net and D-Net, and combining with either or both of the embedding loss $\mathcal{L}_{\rm e}$ and discrimination loss $\mathcal{L}_{\rm d}$. The four variants are used to compare with each other. For better understanding of the running speed of each variant, we calculate the inference time per image and the image number recognized by each block of the main network. The results are reported in Table 2. It is clear to see that all Fast-FAR variants require less runing time than the baseline M with comparable face verification accuracy, the accuracy for M + E-Net + $\mathcal{L}_{\rm e}$ + $\mathcal{L}_{\rm d}$ is even slightly higher than that of M on LFW. All the testing images are recognized at the last block for M, while quite a number of the input images are recognized in advance for Fast-FAR variants, this is the reason why the running time for Fast-FAR variants are lower than that of the baseline M. For the variant M + E-Net + $\mathcal{L}_{\rm e}$ + $\mathcal{L}_{\rm d}$, the percentages of the recognized images by the 4 blocks are 8.38%, 21.17%, 44.98%, 25.47%; 11.04%, 20.58%, 45.14%, 23.24%; and 6.21%, 15.15%, 21.58%, 57.06% on the datasets LFW, AgeDB-30 and CFP-FP, respectively. It saves about 14.22%, 20.61%, and 7.84% running time on the three datasets, respectively, depending on how many easy face images provided by the testing datasets. More easy images contained within the dataset, less time is required for Fast-FAR. By comparing the settings M + E-Net + $\mathcal{L}_{\rm e}$ vs. M, and M + E-Net + $\mathcal{L}_{\rm d}$ vs. M, it is easy to conclude that both the embedding loss $\mathcal{L}_{\rm e}$ and the discrimination loss $\mathcal{L}_{\rm d}$ are effectiveness for the improvement of face recognition. However, only using one of these two loss functions, the recognition performance may drops slightly compared with M (except the setting M + E-Net + $\mathcal{L}_{\rm e}$ on AgeDB-30 dataset).

We visualize the feature that output by each block of the main network, and compare them with the converted ones by E-Net. Specifically, we randomly select three face images from the test set and use the pre-trained model to extract the mean features from each of the four blocks for visualization. The results are shown in Figure 2. As can be seen, the features output from the 4 blocks are presenting at different scales (*Col.* A), even for the same identity. However, the scales for the converted features are almost the same, meaning E-Net have the capacity to convert the shallow-level feature to high-level feature with the same scale, so that shallow-block features can be compared with deep-block in the same space.

C. COMPARISON WITH STATE-OF-THE-ARTS

We further compare face verification performance of our Fast-FAR with state-of-the-art face recognition methods. For a fair comparison with the very recently released work ArcFace [10], we use ResNet-100 as the main network the same

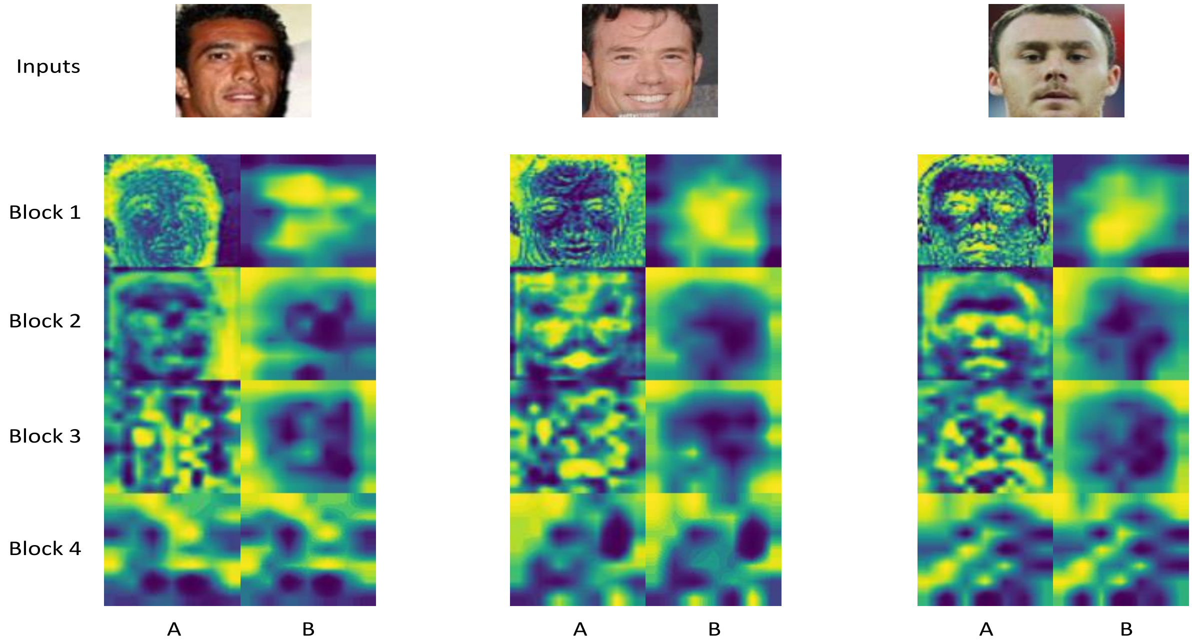

FIGURE 2: Feature visualization results. Results in column "A" are the original features that exacted from the main network, *i.e.,* ResNet-50. Results in column "B" are the converted features by E-Net.

TABLE 3: Face verification performance (%) of different methods on LFW, and AgeDB-30. '-' means the result is not reported.

Method	LFW [17]	AgeDB-30 [19]
DeepID [1]	99.47	-
VGG Face [21]	98.95	-
Softmax [21]	99.08	92.33
Center Loss [6]	99.28	-
SphereFace [8]	99.42	91.70
CosFace [9]	99.51	94.56
ArcFace [10]	99.53	95.15
Fast-FAR + ArcFace Loss	99.58	97.03

with ArcFace, and employ ArcFace loss to train Fast-FAR. The results are shown in Table 3. Other method's results are copied from the paper [10]. As most of the comparing methods have reported their recognition results on LFW and AgeDB-30 while few of then reported the results on CFP-FP, we only use LFW and AgeDB-30 for the comparison of the popular face recognition methods. As can be seen, Fast-FAR with with ResNet-100 beats all the comparison methods by a significant margin on both LFW and AgeDB-30. Specifically, Fast-FAR outperforms the methods DeepID, VGG Face, Softmax, Center Loss, SphereFace, CosFace by 0.11%, 0.63%, 0.5%, 0.3%, 0.16%, 0.07% on LFW dataset, and outperforms Softmax, SphereFace and CosFace by 4.7%, 5.33% and 2.47% on AgeDB-30 dataset. Especially on the comparison with ArcFace which has the same experimental setting, our Fast-FAR can improve the face verification accuracy by 0.05% on LFW dataset, and 1.88% on AgeDB-30 respectively, with faster processing speed. The results indicate that our Fast-FAR can achieve high-speed face recognition without drops recognition accuracy.

IV. CONCLUSION

In this paper, we propose a novel and generic model to speed up face recognition approaches that use Deep Convolutional Neural Networks (DCNN). Based on the observation that most of the easy face images can be well classified by the shallow layers of a DCNN, we train our FAce Recognizer (Fast-FAR) by a manner of reinforcement learning to adaptively learn the earliest layer where the give face image can be accurately recognized. In the experiment, we evaluate our Fast-FAR by comparing with other recognition methods on the popular face recognition benchmarks. The results have demonstrated that Fast-FAR can significantly reduce the recognition time, as well as achieving first-rate face recognition performance. Observing from the experimental results, the performances of our method on some databases are slightly lower than state-of-the-arts. In the future, we will focus on the architecture design of the Embedding sub-Network and the Decision sub-Network, as well as the block partition of the main network, with the goal of further improving the recognition performance on all popular face recognition benchmarks.

REFERENCES

[1] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in Proceedings of the IEEE conference on computer vision and pattern recognition, 1891–1898 (2014).

[2] Y. Taigman, M. Yang, M. Ranzato, et al., "Deepface: Closing the gap to human-level performance in face verification," in Proceedings of the IEEE conference on computer vision and pattern recognition, 1701–1708 (2014).

[3] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2892–2900 (2015).

[4] Y. Sun, D. Liang, X. Wang, et al., "Deepid3: Face recognition with very deep neural networks," arXiv preprint arXiv:1502.00873 (2015).

[5] W. Liu, Y. Wen, Z. Yu, et al., "Large-margin softmax loss for convolutional neural networks.," in ICML, **2**(3), 7 (2016).

[6] Y. Wen, K. Zhang, Z. Li, et al., "A discriminative feature learning approach for deep face recognition," in European conference on computer vision, 499–515, Springer (2016).

[7] X. Tu, J. Gao, M. Xie, J. Qi, Z. Ma, Illumination normalization based on correction of large-scale components for face recognition, Neurocomputing 266 (2017) 465–476.

[8] W. Liu, Y. Wen, Z. Yu, et al., "Sphereface: Deep hypersphere embedding for face recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 212–220 (2017).

[9] H. Wang, Y. Wang, Z. Zhou, et al., "Cosface: Large margin cosine loss for deep face recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5265–5274 (2018).

[10] J. Deng, J. Guo, N. Xue, et al., "Arcface: Additive angular margin loss for deep face recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4690–4699 (2019).

[11] X. Tu, Z. Ma, J. Zhao, G. Du, M. Xie, J. Feng, Learning generalizable and identity-discriminative representations for face anti-spoofing, ACM Transactions on Intelligent Systems and Technology (TIST) 11 (5) (2020) 1–19.

[12] X. Tu, F. Yang, M. Xie, Z. Ma, Illumination normalization for face recognition using energy minimization framework, IEICE TRANSACTIONS on Information and Systems 100 (6) (2017) 1376–1379.

[13] 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, 815–823 (2015).

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

[15] L. De Bortoli, F. Guzzi, S. Marsi, et al., "A fast face recognition cnn obtained by distillation," in International Conference on Applications in Electronics Pervading Industry, Environment and Society, 341–347, Springer (2019).

[16] Y. Guo, L. Zhang, Y. Hu, et al., "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," (2016).

[17] G. B. Huang, M. Mattar, T. Berg, et al., "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," (2008).

[18] X. Tu, J. Zhao, Q. Liu, W. Ai, G. Guo, Z. Li, W. Liu, J. Feng, Joint Face Image Restoration and Frontalization for Recognition, IEEE Transactions on Circuits and Systems for Video Technology, 2021.

[19] S. Moschoglou, A. Papaioannou, C. Sagonas, et al., "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 (2017).

[20] S. Sengupta, J.-C. Chen, C. Castillo, et al., "Frontal to profile face verification in the wild," in 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), 1–9, IEEE (2016).

[21] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," (2015).

[22] Xiaoguang Tu, Jian Zhao, Qiankun Liu, Wenjie Ai, Guodong Guo, Zhifeng Li, Wei Liu, and Jiashi Feng. Joint face image restoration and frontalization for recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[23] Jiankang Deng, Jia Guo, Tongliang Liu, Mingming Gong, and Stefanos Zafeiriou. Sub-center arcface: Boosting face recognition by large-scale noisy web faces. In *European Conference on Computer Vision*, pages 741–757. Springer, 2020.

[24] Jie Chang, Zhonghao Lan, Changmao Cheng, and Yichen Wei. Data uncertainty learning in face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5710–5719, 2020.

[25] Jianzhu Guo, Xiangyu Zhu, Chenxu Zhao, Dong Cao, Zhen Lei, and Stan Z Li. Learning meta face recognition in unseen domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6163–6172, 2020.