Lightweight Face Recognition Challenge

Jiankang Deng [1] Jia Guo [1]
Debing Zhang [2] Yafeng Deng[2] Xiangju Lu [3] Song Shi [3]
[1]InsightFace [2]DeepGlint [3]IQIYI

Abstract

Face representation using Deep Convolutional Neural Network (DCNN) embedding is the method of choice for face recognition. Current state-of-the-art face recognition systems can achieve high accuracy on existing in-the-wild datasets. However, most of these datasets employ quite limited comparisons during the evaluation, which does not simulate a real-world scenario, where extensive comparisons are encountered by a face recognition system. To this end, we propose two large-scale datasets (DeepGlint-Image with 1.8M images and IQIYI-Video with 0.2M videos) and define an extensive comparison metric (trillion-level pairs on the DeepGlint-Image dataset and billion-level pairs on the IQIYI-Video dataset) for an unbiased evaluation of deep face recognition models. To ensure fair comparison during the competition, we define light-model track and large-model track, respectively. Each track has strict constraints on computational complexity and model size. To the best of our knowledge, this is the most comprehensive and unbiased benchmarks for deep face recognition. To facilitate future research, the proposed datasets are released and the online test server is accessible as part of the Lightweight Face Recognition Challenge at the International Conference on Computer Vision, 2019.

1. Introduction

Face recognition in static images and video sequences captured in unconstrained recording conditions is one of the most widely studied topics in computer vision due to its extensive applications in surveillance, law enforcement, bio-metrics, marketing, and so forth.

Recently, great progress has been achieved in face recognition with deep learning-based methods [31, 32, 28, 26, 8, 21, 37, 35, 5] DCNNs map the face image, typically after a pose normalisation step [43, 7], into a feature that should have intra-class compactness and inter-class discrepancy.

Popular evaluation datasets for face recognition includes (1) fast face verification datasets (*e.g*. LFW [16], CFP-FP [29], CPLFW [46], AgeDB-30 [25] and CALFW [47]), (2) large-scale face verification and identification datasets (*e.g*. MegaFace [17], IJB-B [39], and IJB-C [24]), and (3) video-based face verification datasets (*e.g*. YTF [40]). However, most of these datasets employ quite limited comparisons during the evaluation, which does not simulate a real-world scenario, where extensive comparisons are encountered by a face recognition system. In this paper, we propose a very straightforward approach, comparing all possible positive and negative pairs, for a comprehensive evaluation. We collect two large-scale datasets (DeepGlint-Image with 1.8M images and IQIYI-Video with 0.2M videos) and define an extensive comparison metric (trillion-level pairs on the DeepGlint-Image dataset and billion-level pairs on the IQIYI-Video dataset) for an unbiased evaluation of deep face recognition models. The images and videos are collected from the Internet, resulting in unconstrained face images/frames similar to real-world settings.

Even though comprehensive benchmarks exist for deep face recognition, very limited effort has been made towards benchmarking lightweight deep face recognition, which aims at model compactness and computation efficiency to enable efficient system deployment. In this paper, we make a significant step further and propose a new comprehensive benchmark. We define light-model track and large-model track, thus performance comparison between different models can be fairer. Each track has strict constraints on computational complexity and model size. The light-model track targets on face recognition on embedded systems with 1 GFLOPs upper bound of the computation complexity, while the large-model track targets on face recognition on cloud systems with 30 GFLOPs upper bound.

By using the DeepGlint-Image dataset and the IQIYI-Video dataset, we have organised a Lightweight Face Recognition Challenge (ICCV 2019) with four different tracks. This paper presents the benchmarks in detail, including the evaluation protocols, baseline results, perfor-

InsightFace is a nonprofit Github project for 2D and 3D face analysis. Image and video test data are provided by DeepGlint and IQIYI, respectively.

mance analysis of the top-ranked submissions received as part of the competition, challenge cases analysis within the large-scale image and video face recognition, and effective strategies for deep face recognition.

2. Datasets

The following subsections present the dataset statistics of the lightweight face recognition challenge. The pre-processed training and testing datasets are publicly available for research purposes and can be downloaded from our website.

2.1. Training Dataset

Our training dataset is cleaned from the MS1M [9] dataset. All face images are pre-processed to the size of 112×112 by the five facial landmarks predicted by RetinaFace [6]. Then, we conduct a semi-automatic refinement by employing the pre-trained ArcFace [5] model and ethnicity-specific annotators. Finally, we get a refined MS1M dataset named MS1M-RetinaFace, which contains 5.1M images of 93K identities. The training data is fixed to facilitate future comparison and reproduction. Detailed requirements:

- All participants must use this dataset for training without any modification (*e.g*. re-alignment or changing image size are both prohibited).
- Any external dataset is not allowed.

2.2. Large-scale Image Test Set

We take the DeepGlint-Image dataset as our large-scale image test set. The DeepGlint-Image dataset consists of the following two parts:

- ELFW: Face images of celebrities in the LFW [16] name list. There are 274K images from 5.7K identities.
- DELFW: Distractors for ELFW. There are 1.58M face images from Flickr.

All test images are pre-processed to the size of 112×112 (same as the training data). Modification (*e.g*. re-alignment or resize) on test images is not allowed. Horizontal flipping is allowed for test augmentation while all other test argumentation methods are prohibited. The multi-model ensemble strategy is also not allowed.

2.3. Large-scale Video Test Set

We take the IQIYI-Video dataset as our large-scale video test set. The IQIYI-Video dataset is collected from IQIYI variety shows, films and television dramas. The length of each video ranges from 1 to 30 seconds. The IQIYI-Video dataset includes 200K videos of 10K identities.

Dataset	# ID	# Image/frame
MS1M-RetinaFace	93K	5.1M
DeepGlint-Image	5.7K	274K+1.58M(Distractors)
IQIYI-Video	10K	6.3M(from 200K videos)

Table 1. Statistics of the training and testing sets of the lightweight face recognition challenge.

Face frames are extracted from each video at 8FPS and pre-processed to the size of 112×112 (same as the training data). We provide 6.3M pre-processed face crops instead of original videos to simplify the competition. The mapping between videos and frames are also provided and participants can investigate how to aggregate frame features to video feature. Modification (*e.g*. re-alignment or resize) on test images is not allowed. Horizontal flipping is allowed for test augmentation while all other test argumentation methods are prohibited. The multi-model ensemble strategy is also not allowed.

2.4. Dataset Statistics

Dataset statistics of the lightweight face recognition challenge are presented in Table 1. MS1M-RetinaFace is used as the training dataset, while DeepGlint-Image and IQIYI-Video are employed as the large-scale image and video test datasets, respectively.

3. Evaluation Protocols

The lightweight face recognition challenge has four protocols for evaluation. All four protocols correspond to 1:1 verification protocols, where a face recognition model is expected to classify a pair of images/videos as positive or negative pair. More specifically, we choose TPR@FPR as our evaluation metric. A detailed description of each protocol with different constraints on the computational complexity is given below:

- Protocol-1 (DeepGlint-Light) evaluates a lightweight face recognition model for its ability to distinguish image pairs with high precision (FPR@1e-8).
- Protocol-2 (DeepGlint-Large) evaluates a large face recognition model for its ability to distinguish image pairs with high precision (FPR@1e-8).
- Protocol-3 (IQIYI-Light) evaluates a lightweight face recognition model for its ability to distinguish video pairs with high precision (FPR@1e-4).
- Protocol-4 (IQIYI-Large) evaluates a large face recognition model for its ability to distinguish video pairs with high precision (FPR@1e-4).

3.1. Light Model Constraints

In the light model track, we refer the application scenario of unlocking mobile telephone with smooth user experience ($<$ 50ms on ARM). Detailed requirements:

Dataset	# Positive	# Negative
DeepGlint-Image	11,039,533	330,145,575,217
IQIYI-Video	1,550,033	14,561,114,695

Table 2. Positive and negative pair numbers within the image and video testing sets of the lightweight face recognition challenge.

- The upper bound of computational complexity is 1G FLOPs.
- The upper bound of model size is 20MB.
- We target on float32 solutions. Float16, int8 or any other quantization methods are not allowed.
- The upper bound of the feature dimension is 512.

3.2. Large Model Constraints

In the large model track, we refer to the submission requirement of the face recognition vendor test ($<$ 1s on CPU). Detailed requirements:

- The upper bound of computational complexity is 30G FLOPs.
- We target on float32 solutions. Float16, int8 or any other quantization methods are not allowed.
- The upper bound of the feature dimension is 512.

3.3. Pair Statistics

To correctly follow the above-mentioned protocol and report the corresponding accuracy, we use a mask matrix to extract relevant positive and negative pairs. In Table 2, we give the positive and negative pair numbers within the image and video testing sets. We believe extensive pair comparison (*e.g*. trillion-level for images and billion-level for videos) can provide an unbiased evaluation for the face recognition models.

3.4. Submission Format

We have released an online test server for efficient evaluations. For the DeepGlint-Image test set, the participants need to submit a binary feature matrix (ImageNum $\times$ FeatureDim in float32) to the test server. For the IQIYI-Video test set, the participants also need to submit a binary feature matrix (VideoNum $\times$ FeatureDim in float32) to the test server.

4. Baseline Solutions

Baseline models are released before the challenge to facilitate participation. We customise the MobileNet [13] for the light baseline model and the ResNet [10] for the large baseline model. We employ ArcFace [5] as our loss function, which is one of the top-performing methods for deep face recognition.

Light Baseline Model. The detailed network configuration of our light baseline model is summarised in Table 3. The computational complexity is 1.0G FLOPS and the model size is 19.80MB.

Input	**Operator**	t	c	n	s
$112^2 \times 3$	Conv3 $\times$ 3	-	32	1	2
$56^2 \times 32$	Depthwise Conv3 $\times$ 3	1	64	3	2
$28^2 \times 64$	Depthwise Conv	2	64	10	1
$28^2 \times 64$	Depthwise Conv	2	128	17	2
$14^2 \times 128$	Depthwise Conv	4	128	5	2
$7^2 \times 128$	Depthwise Conv	2	128	1	1
$7^2 \times 128$	Conv3 $\times$ 3	-	512	1	2
$4^2 \times 512$	FC	-	256	-	-

Table 3. The network configuration of our light baseline model. Each line represents a sequence of identical layers, repeating n times. All layers in the same sequence have the same number c of output channels. The first layer of each sequence has a stride s. The expansion factor t is always applied to the input size.

Large Baseline Model. As shown in Table 4, we use ResNet124 [10, 5] as our large baseline model. Compared to ResNet100, the block setting is changed to (3, 13, 40, 5), making model deeper. The computational complexity is 29.70G FLOPS and the model size is 297MB.

layer name	124-layer	output size
Input Image Crop		112×112×3
	3×3, 64, stride 1	112×112×64
Conv2_x	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	56×56×64
Conv3_x	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times13$	28×28×128
Conv4_x	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times40$	14×14×256
Conv5_x	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times5$	7×7×512
FC		1×1×512

Table 4. The network configuration of our large baseline model. Convolutional building blocks are shown in brackets with the numbers of blocks stacked. Down-sampling is performed by the second conv in conv2_1, conv3_1, conv4_1, and conv5_1 with a stride of 2.

4.1. Implementation Details

During training, we follow [5] to set the feature scale to 64 and choose the angular margin of ArcFace at 0.5. The

baseline models are implemented by MXNet [3] with parallel acceleration on both features and centres[1]. We set the batch size to 512 and train models on four NVIDIA Tesla P40 (24GB) GPUs. We divide the learning rate at 100K,160K,220K iterations and finish at 240K iterations. We set the momentum to 0.9 and weight decay to $5e-4$. During testing, we only keep the feature embedding network without the fully connected layer and extract the 512-D features for each normalised face crop. To get the embedding features for videos, we simply calculate the feature centre of all frames from the video. Flip testing is used in our baseline models by addition and then normalisation. The extracted features are compared using Cosine distance, followed by a threshold to distinguish positive or negative pairs.

5. Top-ranked Competition Solutions

The lightweight face recognition competition is conducted as part of the *Lightweight Face Recognition Challenge & Workshop*[2], at the International Conference on Computer Vision 2019 (ICCV 2019). All participating teams are provided with the training and the testing datasets. Participants are required to develop a face feature embedding algorithm, which is automatically evaluated on our test server based on the above-mentioned four protocols.

The competition has been opened worldwide, to both industry and academic institutions. The competition has received 292 registrations from across the world. More specifically, the competition has received 112 submissions for the DeepGlint-Light track, 91 submissions for the DeepGlint-Large track, 53 submissions for the IQIYI-Light track, and 45 submissions for the IQIYI-Large track. Here, multi-submissions for one protocol from the same participant is only counted for once. After the competition, we close the test server and select the valid top-3 solutions for each protocol. We collect the training code from these top-ranked participants and re-train the models to confirm (1) whether the performance of each submission is valid or not, and (2) whether the computational complexity of each submission is within requirement or not.

Table 5 presents a list of the top-ranked participating teams from all over the world, having both industry and academic affiliations. Details regarding the technique applied by each submission are provided below:

YMJ for DeepGlint-Light: "YMJ" is a submission [41] from an anonymous affiliation. Their solution is named VarGFaceNet, which employed variable group convolution [44] to reduce computational cost and parameter number. More specifically, they use a head setting to reserve essential information at the beginning of the network and propose a particular embedding setting to reduce parameters of fully-connected layer for embedding. To enhance interpretation ability, they employe an equivalence of angular distillation loss to guide the lightweight network and apply recursive knowledge distillation to relieve the discrepancy between the teacher model and the student model.

count for DeepGlint-Light: "count" is a submission [19] from AIRIA. Their solution is named AirFace, which has proposed a novel loss function named Li-ArcFace based on ArcFace. Li-ArcFace takes the value of the angle through linear function as the target logit rather than through cosine function, which has better convergence and performance on low dimensional embedding feature learning for face recognition. In terms of network architecture, they improve the performance of MobileFaceNet [2] by increasing the network depth and width and adding attention module.

NothingLC for DeepGlint-Light: "NothingLC" is a submission from MSRA. They used a teacher-student framework for the lightweight face recognition task. Firstly, they train a DenseNet [15] as the teacher model. Then, they directly copy and fix the weights of the margin inner-product layer to the student model to train student model from scratch. In this way, the student model can be trained with better pre-defined inter-class information from the teacher model. For the backbone, they select a modified version of the ProxylessNAS mobile network [1]. In detail, they use PReLU to take place of ReLU as the activation function, replace the last global average pooling layer with global depth-wise convolution layer [2], add SE layers [14] with a reduction ratio of 4, and scale up the width to make the model larger. The loss function they used is AMSoftmax [35], with scale at 50 and margin at 0.45.

lhlh18 for DeepGlint-Large: "lhlh18" is a submission from CAS Institute of Automation (CASIA). They have designed a modified version of residual attention network [34] for the backbone and applied CosFace [37] as the loss function. Based on Attention-56 [34], they adjust the input size, increase the number of Attention Modules in each phase, and change the output layer from average pooling to BN-Dropout-FC-BN [5]. In the training process, they first use softmax loss to train the network from scratch. The learning rate starts from 0.1 and is divide by 10 at 3, 6, 9 epochs. The total epoch number is 12. After the softmax loss converges, they use CosFace [37] instead. The learning rate starts from 0.01 and is divide by 10 at 3, 6, 9, 12 epochs. The CosFace training process has 15 epochs. The margin in CosFace is set to 0.48. They use two TITANXp GPUs for training and the batch size is 62.

tiandu for DeepGlint-Large: "tiandu" is a submission from JD AI Lab. They have developed a new architecture named AttentionNet-IRSE and proposed a three-stage training strategy. They integrate the AttentionNet [34] and the IRSE module [5] into one framework. The depth stages of

[1] https://github.com/deepinsight/insightface/tree/master/recognition

[2] https://ibug.doc.ic.ac.uk/resources/lightweight-face-recognition-challenge-workshop/

Participant Name	Affiliation	Brief Description
YMJ	-	variable group convolution, angular distillation loss, recursive knowledge distillation
count	AIRIA	improved MobileFaceNet, Li-ArcFace
NothingLC	MSRA	modified ProxylessNAS mobile, AMSoftmax, knowledge distillation
lhlh18	CASIA	modified residual attention network, two-stage training (Softmax, CosFace)
tiandu	JD AI Lab	AttentionNet-IRSE, three-stage training (NSoftmax,ArcFace,AMSoftmax+MHE)
dengqili	Bytedance AI Lab	multi-path ResNet100, combined loss
NothingLC	MSRA	modified ProxylessNAS mobile, AMSoftmax, knowledge distillation, quality-aware aggregation
Rhapsody	-	EfficientNet, three-stage training (distillation, ArcFace, adapt-fusion)
xfr	NetEase Games AI Lab	improved MobileFaceNet, two-stage training (NSoftmax, ArcFace + SVX)
trojans	CUHK and Sensetime	Efficient PolyFace, adj-Arcface, quality aware network++
NothingLC	MSRA	DenseNet290, AMSoftmax, quality-aware aggregation
trantor	Alibaba-VAG	ResNetSE-152, CosFace
PES	Pensees	off-line graph-based unsupervised feature aggregation

Table 5. List of top-ranked teams which participated in the lightweight face recognition challenge.

AttentionNet-IRSE [38] are set to 3,6,2. In the first training stage, the N-Softmax loss [36] is used to train the model. The scale parameter is set as 32. The learning rate is initialised at 0.1 and divided by 10 at 3, 6, 9, and 11 epochs, finishing at 12 epochs. In the second training stage, ArcFace [5] is used to fine-tune the model from the first stage. The scale and the margin of ArcFace are set to 64 and 0.5, respectively. The initial learning rate is 5e-3 and divided by 10 at 4, 7, 10 epochs, finishing at 12 epochs. In the third training stage, the AM-Softmax loss [35] with the MHE [20] regularisation on the last fully connected layer is used to further fine-tune the model. The initial learning rate is 5e-4, and the scale and the margin of AM-Softmax are 32 and 0.45, respectively. The learning rate is reduced at 4, 8 epochs and the maximal epoch is 9.

dengqili for DeepGlint-Large: "dengqili" is a submission from Bytedance AI Lab. They proposed a framework that fuses features from multiple face patches(*i.e*. one global face patch and four local face patches). The global patch can obtain global features, while the local patch can obtain more details. Based on these observations, the feature maps of Conv3 block in ResNet100 [10] are cropped into four 16×16 patches at the location of (4, 4), (8, 8), (4, 8), and (8, 4), and then these crops are feeded into four sub-nets individually to learn local features. There are three stages in the sub-net: subnet-res3-ex, subnet-res4 and subnet-res5, and the number of channels and block are (128,3), (256,9) and (512,3), respectively. Finally, the features from the main network and four sub-nets are fused by element-wise addition. Combined loss [5] is used to train the network.

NothingLC for IQIYI-Light: "NothingLC" is a submission from MSRA. They have used a knowledge distillation method to guide the light student model by the large teacher model and aggregated the features from different video frames by a quality-aware method. The teacher network is a DenseNet and the student network is a modified version of ProxylessNAS mobile network. Last margin inner-product layer in the teacher network is copied to the student model and fixed during training. The loss function contains two parts: AMSoftmax loss and L_2 loss between teachers embedding feature and students embedding feature. The weight of the L_2 loss is set to 0 in the first epoch, then set to 1 in the following 100 epochs, and finally set to 100 for another 10 epochs. For the AMSoftmax loss, the scale is set to 60 and the margin is set to 0.35. For the feature aggregation from video frames, the cubic of feature norm is used as the frame-wise quality weight.

Rhapsody for IQIYI-Light: "Rhapsody" is a submission from an anonymous affiliation. There are three stages involved in the training process. In the first stage, they use a pre-trained ResNet100 [5] as the teacher network to guide the training of the light-weight student network, which known as knowledge transfer. The backbone of the student network is based on EfficientNet-b0 [33], but with several modifications: (1) the input size is 112×112, and (2) the stride in first conv-block is changed to 1, and (3) the width is set to 1.1. The feature dimension for both teacher and student is 256. The student network is trained by minimising (1) L_2 regression loss [18] between features from the teacher and student networks, and (2) the KL loss [12] between part final predictions of teacher and student [27]. The weight of L_2 regression loss is 1.0, and the scale of the KL loss is 0.1. To better transfer the knowledge of the teacher into the student, they propose a selective knowledge distillation based on the confidence of the teachers prediction. In the second stage, the hard label information is used to fine-tune the model derived from stage one to further improve the student network. ArcFace loss is used here, but with a much smaller initial learning rate. In the final stage, they add an adapt-fusion component on the student network. There are two goals of employing adapt-fusion component: (1) domain adaptation for better extract feature on each frame from the video, and (2) feature fusion based on attention to aggregate features of multi-frames into

video-level representation. The adapt-fusion is consisted of a fully connected layer (256×256) and an attention block (256×1 as in NAN [42]). During training in the final stage, they fix the parameters of the backbone (EfficientNet) and only train the adapt-fusion block with ArcFace.

xfr for IQIYI-Light: "xfr" is a submission from NetEase Games AI Lab. They employed a narrower and deeper version of MobileFaceNet [2]. In detail, the filter number of the first convolution layer is set to 32 and the output size of the first block is set to 32. The output size of the third block is set to 96, and the output size of the final block is set to 256. Meanwhile, the block setting is changed to (2, 8, 22, 20) from (2, 8, 16, 4) to make model deeper. To make the training process more stable, NSoftmax [36] is used for a few epochs and then Arcface [5] and SVX [38] are used to get the final model. As the magnitude of the feature is highly related to the quality of the input face, L_2 normalisation is not directly used on the frame-wise feature. They compute the weighted average of the extracted features, using the cube of the norm of each feature as the weight. Finally, L_2 normalisation is conducted on the aggregated feature.

trojans for IQIYI-Large: "trojans" is a submission [22] from CUHK and Sensetime. They have employed a network architecture named Efficient PolyFace, a new loss function named adj-Arcface, and a novel frame aggregation method named QAN++. Inspired by the idea of efficientnet [33], they launch a NAS processing to expand the basic PolyNet [45] models in depth and width with the constraint of the computation budget. After the network architecture search, they find one of the Efficient PolyFace models outperforms all searched candidates with the same FLOPs. For the loss function, they not only use additive angular margin penalty on the target logit like ArcFace [5], but also add another adaptive adjustment on other inter-class cosine distances. Inspired by QAN and RQEN [23, 30], they propose a new quality estimation strategy called QAN++, which assigns the image quality from the characteristics of feature discrimination. Finally, the feature of the video can be aggregated by the weighted sum with the assistant of the predicted image quality.

NothingLC for IQIYI-Large: "NothingLC" is a submission from MSRA. They have used the DenseNet-290, a much deeper modification compared to the official DenseNet-161, as the backbone. K factor is set to 48. Global average pooling is replaced by a fully-connected layer for feature extraction with dropout setting (0.3). AM-Softmax is used as the loss function with the scale at 60 and the margin at 0.35. For quality-aware aggregation, feature norm is used as the quality factor. The cubic of quality is used to weigh different frame-wise features to get the final video feature representation.

trantor for IQIYI-Large: "trantor" is a submission from Alibaba-VAG. The backbone of the network is ResNet-152 [10] with Squeeze-and-Excitation blocks [14]. For the down-sampling block, a 2×2 average pooling layer with a stride of 2 is used before the convolution [11]. The training loss is CosFace [37] with the margin at 0.48. Since the IQIYI dataset has lots of low-quality frames, the L2 normalisation on the frame-wise feature is removed during testing to improve the performance.

PES for all tracks: "PES" is a submission [4] from Pensees. They innovatively propose a graph-based unsupervised feature aggregation method to directly improve the ROC curve. This method uses the similarity scores between pairs and refines the pair-wise scores to achieve intra-class compactness during testing. First, based on the assumption that all face features follow Gaussian distribution, they derive an iterative updating formula of features. Second, in discrete conditions, they build a directed graph where the affinity matrix is obtained from pair-wise similarities and filtered by a pre-defined threshold along with K-nearest neighbour. Third, the affinity matrix is used to obtain a pseudo centre matrix for the iterative update process. Since this method is a post-processing off-line method, we set a separate track for this method instead of directly compare it with above-mentioned methods.

6. Results

6.1. Competition Results

Tables 6-9 report True Possitive Rate (TPR) at different False Positive Rates (FPRs). Figure 1 and 2 present the Receiver Operating Characteristic (ROC) curves of the above mentioned models. Results for each track are given as below:

Results on DeepGlint-Light: Table 6 presents the TPR corresponding to different FPR values, and Figure 1(a) presents the ROC curves. It can be observed that for this track, "YMJ" outperforms other algorithms by achieving 88.78% at FPR=1e-8. The second and third methods are from "count" and "NothingLC", which present a verification accuracy of 88.42% and 88.14%, respectively. Compared to the baseline result (84.02%), the solutions from the competition improve the TPR by more than 4%. Even though the offline solution from "PES" can significantly improve TPR at FPR=1e-8, there is an obvious performance drop at more strict FPR.

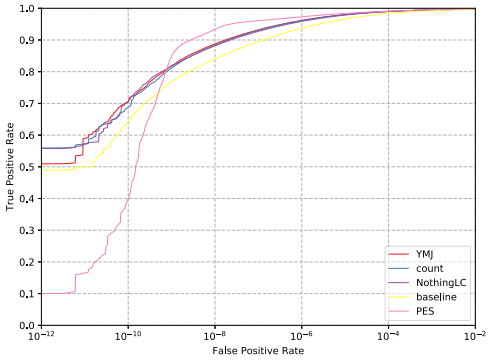
Results on DeepGlint-Large: Table 7 summarises the TPR corresponding to different FPR values, and Figure 1(b) presents the ROC curves. "lhlh18" achieves the best accuracy of 94.19%. The second and third methods are "'tiandu' and "dengqili", which achieve verification accuracy of 93.97% and 93.94%, respectively. Compared to the baseline performance (93.37%), the competition solutions only improve TPR by less than 1%. By using AdaBN, the performance of "trojans" improves from 93.81% to 94.20%.

Participants	1e-11	1e-10	1e-09	**1e-08**	1e-07	1e-06	1e-05	1e-04	1e-03	1e-02	1e-01
YMJ[1]	0.5898	0.7060	0.8189	**0.8878**	0.9323	0.9620	0.9803	0.9900	0.9947	0.9971	0.9989
count[2]	0.5728	0.6888	0.8120	**0.8842**	0.9305	0.9614	0.9801	0.9899	0.9946	0.9970	0.9989
NothingLC[3]	0.5712	0.7050	0.8179	**0.8814**	0.9273	0.9591	0.9787	0.9893	0.9945	0.9973	0.9992
baseline	0.5000	0.6442	0.7670	**0.8402**	0.8953	0.9375	0.9664	0.9835	0.9923	0.9966	0.9990
PES	0.1650	0.3996	0.8499	0.9341	0.9610	0.9736	0.9838	0.9909	0.9956	0.9983	0.9996

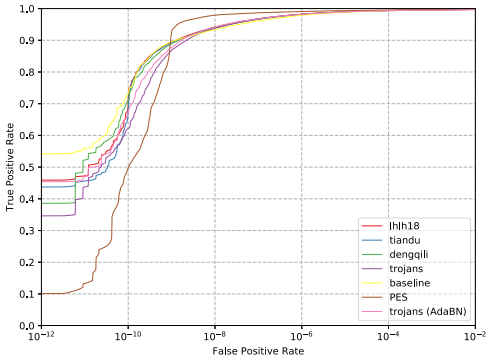
Table 6. Verification accuracy of top-ranked participants and baseline in the DeepGlint-Light track.

Participants	1e-11	1e-10	1e-09	**1e-08**	1e-07	1e-06	1e-05	1e-04	1e-03	1e-02	1e-01
lhlh18[1]	0.4723	0.7231	0.8945	**0.9419**	0.9681	0.9826	0.9902	0.9939	0.9958	0.9972	0.9988
tiandu[2]	0.4564	0.6915	0.8928	**0.9397**	0.9667	0.9818	0.9898	0.9938	0.9958	0.9972	0.9988
dengqili[3]	0.5224	0.7113	0.8884	**0.9394**	0.9664	0.9817	0.9899	0.9940	0.9960	0.9974	0.9989
trojans	0.4347	0.6221	0.8695	**0.9381**	0.9669	0.9819	0.9897	0.9934	0.9954	0.9968	0.9984
baseline	0.5592	0.7336	0.8958	**0.9337**	0.9614	0.9788	0.9885	0.9934	0.9958	0.9974	0.9989
PES	0.1332	0.4973	0.9282	**0.9793**	0.9867	0.9908	0.9936	0.9954	0.9966	0.9981	0.9995
trojans(AdaBN)	0.4701	0.6749	0.8772	**0.9420**	0.9680	0.9821	0.9896	0.9933	0.9953	0.9968	0.9984

Table 7. Verification accuracy of top-ranked participants and baseline in the DeepGlint-Large track.

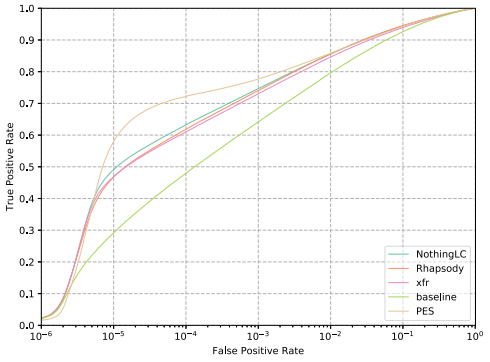


(a) DeepGlint-Light

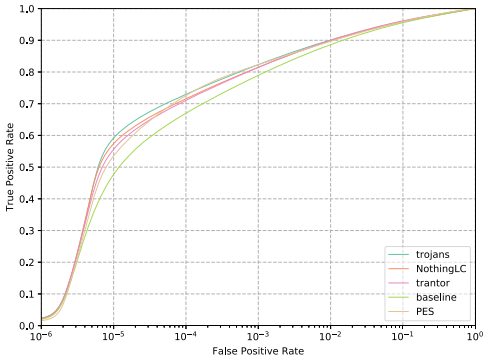


(b) DeepGlint-Large

Figure 1. ROC curves on the DeepGlint dataset.



(a) IQIYI-Light



(b) IQIYI-Large

Figure 2. ROC curves on the IQIYI dataset.

Participants	1e-06	1e-05	**1e-04**	1e-03	1e-02	1e-01
NothingLC[1]	0.0222	0.4915	**0.6323**	0.7471	0.8560	0.9444
Rhapsody[2]	0.0232	0.4678	**0.6187**	0.7412	0.8562	0.9455
xfr [3]	0.0227	0.4695	**0.6105**	0.7303	0.8461	0.9391
baseline	0.0221	0.2923	**0.4803**	0.6417	0.7962	0.9261
PES	0.0160	0.5804	**0.7223**	0.7768	0.8586	0.9436

Table 8. Verification accuracy of top-ranked participants and baseline in the IQIYI-Light track.

Participants	1e-06	1e-05	**1e-04**	1e-03	1e-02	1e-01
trojans[1]	0.0232	0.5921	**0.7298**	0.8231	0.9013	0.9579
NothingLC[2]	0.0227	0.5744	**0.7159**	0.8147	0.8988	0.9602
trantor[3]	0.0231	0.5567	**0.7110**	0.8148	0.9003	0.9609
baseline	0.0200	0.4774	**0.6700**	0.7895	0.8863	0.9551
PES	0.0156	0.5356	**0.7259**	0.8235	0.8960	0.9594

Table 9. Verification accuracy of the participants and baseline in the IQIYI-Large track.

However, using the statistic information from the test set is viewed as an offline method in our challenge. In this track, the offline solution from "PES" significantly improves TPR at FPR=1e-8 once again.

Results on IQIYI-Light: Table 8 illustrates the TPR corresponding to different FPR values, and Figure 2(a) presents the ROC curves. "NothingLC" achieves the best performance of 63.23%. The second and third methods are "Rhapsody" and "xfr", which obtain verification accuracy of 61.87% and 61.05%, respectively. In this track, the performance of our baseline is much worse than the competition solutions. Once again, the offline solution from "PES" significantly boosts TPR at FPR=1e-4.

Results on IQIYI-Large: Table 9 illustrates the TPR corresponding to different FPR values, and Figure 2(b) presents the ROC curves. "trojans" achieves the best accuracy of 72.98%. The second and third methods are "NothingLC" and "trantor", which obtain verification accuracy of 71.59% and 71.10%, respectively. Compared to the baseline result (67.00%), the solutions from the competition obviously improve the TPR by around 6%. The offline solution from "PES" also obtains high TPR at FPR=1e-4.

7. Conclusions and Future Works

In this paper, we introduce our new benchmark for the evaluation of both image-based and video-based deep face recognition with different constraints on the computational complexity. After the analysis of the top-ranked submissions received by the competition, we draw the following conclusions: (1) margin-based softmax loss is the most effective loss function for deep face recognition by now, (2) knowledge distillation is effective for training lightweight models, (3) performance gains from exploring different networks are obvious for lightweight model but not very significant for large model, and (4) quality-aware aggregation is useful to improve video-based face recognition. After the extensive exploration of the network and loss designs, we will try pruning and quantization for lightweight face recognition in the future.

References

[1] H. Cai, L. Zhu, and S. Han. Proxylessnas: Direct neural architecture search on target task and hardware. *arXiv:1812.00332*, 2018.

[2] S. Chen, Y. Liu, X. Gao, and Z. Han. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In *Chinese Conference on Biometric Recognition*, pages 428–438. Springer, 2018.

[3] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv:1512.01274*, 2015.

[4] Y. Cheng, Y. Li, Q. Liu, Y. Yuan, V. S. V. K. Pedapudi, X. Fan, C. Su, and S. Shen. A graph based unsupervised feature aggregation for face recognition. In *ICCV workshops*, 2019.

[5] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

[6] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *arXiv:1905.00641*, 2019.

[7] J. Deng, A. Roussos, G. Chrysos, E. Ververas, I. Kotsia, J. Shen, and S. Zafeiriou. The menpo benchmark for multi-pose 2d and 3d facial landmark localisation and tracking. *International Journal of Computer Vision*, 127(6-7):599–624, 2019.

[8] J. Deng, Y. Zhou, and S. Zafeiriou. Marginal loss for deep face recognition. In *CVPR Workshop*, 2017.

[9] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV*, 2016.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[11] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li. Bag of tricks for image classification with convolutional neural networks. In *CVPR*, 2019.

[12] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.

[13] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*, 2017.

[14] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[16] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, 2007.

[17] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *CVPR*, 2016.

[18] Q. Li, S. Jin, and J. Yan. Mimicking very efficient network for object detection. In *CVPR*, 2017.

[19] X. Li. Airface:lightweight and efficient model for face recognition. In *ICCV workshops*, 2019.

[20] W. Liu, R. Lin, Z. Liu, L. Liu, Z. Yu, B. Dai, and L. Song. Learning towards minimum hyperspherical energy. In *NeurIPS*, 2018.

[21] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017.

[22] Y. Liu, G. Song, M. Zhang, J. Liu, Y. Zhou, and J. Yan. Enhanced quality aware network for video based face recognition. In *ICCV workshops*, 2019.

[23] Y. Liu, J. Yan, and W. Ouyang. Quality aware network for set to set recognition. In *CVPR*, 2017.

[24] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, and J. Cheney. Iarpa janus benchmark–c: Face dataset and protocol. In *ICB*, 2018.

[25] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou. Agedb: The first manually collected in-the-wild age database. In *CVPR Workshop*, 2017.

[26] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015.

[27] B. Peng, X. Jin, K. Yuan, X. Li, S. Zhou, D. Liang, Z. Zhang, J. Liu, and D. Li. Teacher is not oracle! selective knowledge distillation to improve student. 2019.

[28] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[29] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs. Frontal to profile face verification in the wild. In *WACV*, 2016.

[30] G. Song, B. Leng, Y. Liu, C. Hetang, and S. Cai. Region-based quality estimation network for large-scale person re-identification. In *AAAI*, 2018.

[31] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NeurIPS*, 2014.

[32] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

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

[34] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In *CVPR*, 2017.

[35] F. Wang, W. Liu, H. Liu, and J. Cheng. Additive margin softmax for face verification. *SPL*, 2018.

[36] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. Normface: l_2 hypersphere embedding for face verification. *arXiv:1704.06369*, 2017.

[37] H. Wang, Y. Wang, Z. Zhou, X. Ji, Z. Li, D. Gong, J. Zhou, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, 2018.

[38] X. Wang, S. Wang, S. Zhang, T. Fu, H. Shi, and T. Mei. Support vector guided softmax loss for face recognition. *arXiv:1812.11317*, 2018.

[39] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. C. Adams, T. Miller, N. D. Kalka, A. K. Jain, J. A. Duncan, and K. Allen. Iarpa janus benchmark-b face dataset. In *CVPR Workshop*, 2017.

[40] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR*, 2011.

[41] M. Yan, M. Zhao, Z. Xu, Q. Zhang, G. Wang, and Z. Su. Vargfacenet: An efficient variable group convolutional neural network for lightweight face recognition. In *ICCV workshops*, 2019.

[42] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. In *CVPR*, 2017.

[43] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *SPL*, 2016.

[44] Q. Zhang, J. Li, M. Yao, L. Song, H. Zhou, Z. Li, W. Meng, X. Zhang, and G. Wang. Vargnet: Variable group convolutional neural network for efficient embedded computing. *arXiv:1907.05653*, 2019.

[45] X. Zhang, Z. Li, C. Change Loy, and D. Lin. Polynet: A pursuit of structural diversity in very deep networks. In *CVPR*, 2017.

[46] T. Zheng and W. Deng. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. *Technical Report*, 2018.

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