2024 IEEE International Joint Conference on Biometrics (IJCB) | 979-8-3503-6413-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/IJCB62174.2024.10744486

Revisiting Linearization of Spatial Maps in SoTA Face Recognition Backbone

Aman Bhatta Haiyu Wu Kagan Ozturk Kevin W. Bowyer*

University of Notre Dame

{abhatta,hwu6,kozturk,kwb}@nd.edu

Abstract

The prevailing approach in face recognition is to specialize a deep network for general computer vision to the face recognition task. ResNet being specialized for use as the backbone in SoTA face recognition systems is a prime example of this. One significant architectural deviation in the ResNet backbone adapted for face recognition is the linearization of the output spatial map from the last convolution layer to feed the linear layer, rather than utilizing Global Average Pooling (GAP). The utilization of GAP treats all pixel values in the output spatial map as equally significant and averages them naively, thereby compromising the performance of the face recognition model. However, linearization of the spatial map inflates the total parameters in the model by up to 58% (R34) in the lighter version of the ResNet backbone that is typically used for face recognition. Leveraging the prior knowledge that face images during training and testing are pre-aligned, we introduce a novel Gaussian Weighted Pooling (GWP) layer, integrating a pre-computed Gaussian Attention Kernel with the Average Pooling Layer that weighs the importance of the pixel based on the spatial position. Our findings show that utilizing GWP consistently outperforms GAP and achieves results comparable to those of parameter-inflated baseline models.

1. Introduction

ResNet [14] has proven to be a highly successful backbone for various vision tasks, such as classification, detection, and segmentation. Face recognition is one of the fields that has benefited from ResNet's effectiveness as a backbone for network architectures. Beginning with FaceNet [28], which employed deep inception networks for face recognition, subsequent developments continued along a similar trajectory. SphereFace [19], introduced thereafter, also utilized a deep network architecture. However, the seminal work [10] leveraged a variant of the adapted ResNet architecture. Following this breakthrough, recent advancements such as Q-MagFace [33], MagFace [24], AdaFace [18], UniformFace [11], CurricularFace [17], UniFace [42], and RegularFace [39] have all adopted the same backbone initially proposed by [10]. These methods have consistently demonstrated state-of-the-art accuracies across standard academic benchmarks such as LFW [16], CFP-FP [29], AGEDB-30 [25], CALFW [41], CPLFW [40], and the IJB family testing suite [23, 34].

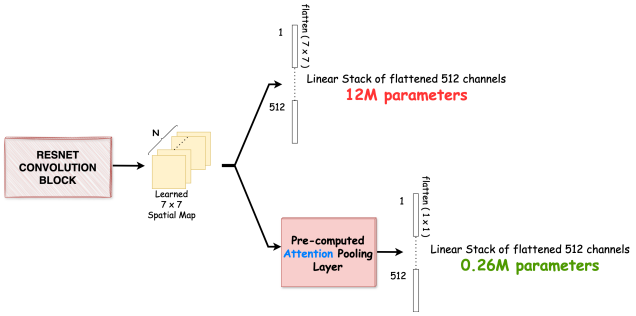


Figure 1. Overview of the proposed Gaussian Pooling Layer (GWP) integrated into a ResNet-based deep face recognition network. Current practice involves linearizing the output spatial map from the last convolution layer, adding significant parameters to the network. GWP offers an alternative approach by precomputing attention pooling, eliminating the need for spatial map linearization, all while having accuracy comparable to parameter-inflated models.

To train a face recognition network, the ResNet backbone is trained with the classification head attached. However, during inference time, the classification head is removed to obtain 512-D feature vectors, which are used as the representation of the image. This 512-D feature vector representation of the image is later used for face verification and identification protocols. This is different from general classification, where the logits are transformed into probability scores using the softmax function, and the index with the highest probability output corresponds to the result class

*Dr. Bowyer is a member of the FaceTec (facetec.com) Advisory Board. Results in this paper do not necessarily relate to FaceTec products.

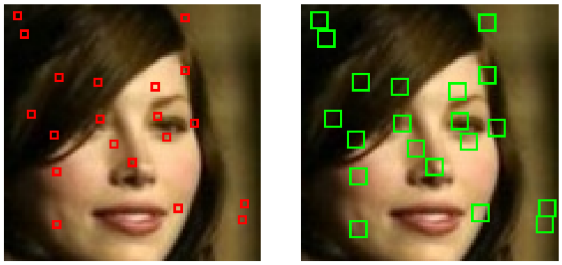
Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on May 21,2025 at 02:58:53 UTC from IEEE Xplore. Restrictions apply.

for a given image. In general vision networks, it's typical for the convolutional output to be passed through an Global Average Pooling (GAP) layer, which serves to condense the spatial map down to a 1×1 size. It's worth noting that in general vision tasks, typically only one linear layer is employed. This layer acts similarly to a basic linear layer in a multi-layer perceptron (MLP), linking the flattened output of the average pooling to the output layer, where each nodes represent the classes. However, in face recognition networks, there are two layers involved. The first one connects the flattened feature map to a 512-D representation, and the second one connects these representations to the nodes representing the number of identities (class). During inference, the last linear layer is removed, and the intermediate 512-D representation is utilized as the image's representation. When referring to the total parameters in the backbone, we exclude the last linear layer (classification head) since its size is solely determined by the number of identities in the training data, akin to the classes in general vision tasks. Our focus lies solely on the backbone, which produces a 512-D feature vector representation for an image.

To ensure a reliable representation of an image in the proxy 512-D vector, numerous modifications have been implemented on the original ResNet architecture. These adaptations have been widely adopted by ResNet-based face recognition approaches. Some of the major architectural adaptations include:

1) **Linearizing the output spatial map after the last convolution layer, as opposed to using Global Average Pooling(GAP)**. The final convolution layer in the SoTA backbone configuration outputs a spatial map of size 7×7 . With 512 feature maps, this flattening of this spatial map adds approximately 13 million ($512 \times 7 \times 7 \times 512$) parameters to the backbone. This alone contributes to approximately 54% of the total parameters in ResNet18, which totals 24 million parameters, and about 20% of the total parameters in ResNet100, which totals 65 million parameters. Conversely, using a Global Average Pooling (GAP) layer would result to addition of only 0.26 million linear ($512 \times 1 \times 1 \times 512$) parameters, or roughly 0.4 - 1% of the total. Efforts to decrease linear parameters have been demonstrated in the field of generative models [27], but such endeavors have been less prevalent in SoTA face recognition networks.

2) **Changing the kernel size from 7×7 and stride 2 followed by pooling to a kernel size of 3×3 and stride 1 with no pooling**. This adjustment preserves a larger spatial map throughout the network. To maintain larger spatial map to learn more distinct features appears to be a logical decision. However, achieving the same spatial output using a 3×3 kernel, stride 1, and padding 1 can also be attained using a 7×7 kernel, stride 1, and padding 3, with negligible additional parameters. *Using a larger kernel in the first layer that interacts with raw RGB image might allow it to potentially model features more effectively as larger kernel extends the receptive field of the network*. One such example of convolution kernels of size 3 and 7 overlaid over the face image is shown in Figure 2. Another motivation for changing the kernel size from 7×7 to 3×3 could stem from VGGNet [30], which attained state-of-the-art results for various computer vision tasks back then. Unlike another popular architecture, Inception [31], which employs a mixture of kernel sizes, VGGNet standardized the convolution kernel size to 3×3 throughout the network. Changing from 7×7 to 3×3 and vice-versa in the first layer has insignificant impact to parameter count, regardless we present the results with both kernel size setting.



(a) k = 3×3 (b) k = 7×7

Figure 2. A 7×7 kernel and a 3×3 kernel are randomly overlaid on the face image. Using a 7×7 kernel directly on the raw RGB image could better model patterns. Additionally, smaller 3×3 kernels might also be susceptible to noise interacting with RAW pixel values. However, employing a smaller kernel size in deeper layers is acceptable as it operates on abstract representations, where noise is essentially filtered out by prior convolution kernels.

3) **Modifications to the residual units and eliminating bottleneck layers**. Residual blocks facilitate effective gradient flow, while bottleneck layers aim to reduce the convolution parameters within the convolution blocks of ResNet. These changes could have a substantial impact for face recognition network. However, the impact of the alternative residual block configuration has been thoroughly studied in this work [10].

The family of lightweight and efficient face recognition networks has garnered attention in recent years [2, 5–8, 13, 21, 35, 36, 38]. This line of research aims to propose various architectures with reduced computational complexity. While other works focus on increasing accuracy numbers on benchmarks using deep neural networks [10, 11, 17–19, 24, 33, 39, 42], our approach combines aspects of both strategies. Rather than proposing an entirely new lightweight network, we analyze one of the

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on May 21,2025 at 02:58:53 UTC from IEEE Xplore. Restrictions apply.

most widely used backbones and identify how a specific layer has been suboptimally integrated and overutilized. The key contributions of our work include:

C1) We introduce a novel Gaussian Weighted Pooling (GWP) layer that addresses the inherent flaw in global average pooling, which discards the spatial information and treats each pixel equally. Furthermore, it achieves accuracy comparable to the original backbone that linearizes the output spatial map, all while significantly reducing the number of parameters .

C2) We have demonstrated the effectiveness of our approach across both static and adaptive margin learning paradigms, as well as across varying depths of face recognition backbones, using standard benchmark datasets and the IJB test family.

2. Literature Review

Face recognition stands as a pivotal application within computer vision. Researchers in this field typically diverge into two primary avenues of exploration. One faction focuses on refining metrics to effectively represent facial features, remaining indifferent to the scale of the underlying backbone. On the other hand, another faction directs its efforts towards optimizing the architecture of the underlying backbone, often prioritizing computational efficiency and scalability over intricate feature representation.

Improving benchmark accuracy. An early milestone in utilizing deep learning for face recognition was achieved by FaceNet [28], which employed an inception network boasting approximately 140 million parameters to generate a 128-dimensional vector for image representation. Subsequently, with the emergence of ResNet [14] , state-of-the-art methodologies adapted versions of ResNet as their backbone [10, 11, 17–19, 24, 33, 39, 42]. These approaches are primarily geared towards maximizing accuracy on benchmark datasets, often overlooking considerations regarding parameter count in the network backbone used. Note that, with the exception of [19], which predates [10], all other advanced deep-learning-based face recognition methods utilize the modified ResNet introduced by [10]. In our study, we draw attention to the unnecessary parameter inflation in the linear layer of the backbones employed by these face recognition methods, which only yield marginal accuracy improvements.

Efficient Face Recognition Networks. Researchers have focused on designing and adapting lightweight face recognition networks to tackle the memory and computational complexities inherent in state-of-the-art deep neural networks. This approach aims to achieve higher benchmark accuracies while mitigating the burdensome resource requirements. One such early approach adapts MobileNet architecture for face [8, 15]. They propose a global-depth wise convolution to treat the final feature maps differently and in doing so, they reduce the numbers of parameters. Similarly, another work called, Efficient Lightweight Attention Networks (ELANet), consists of inverted residual blocks, which can alleviate the computational effort required by fusing spatial and channel attention [38]. Another such work called MixFaceNet [5] uses the concept of MixConv and adapt it to face recognition networks. Mixed depthwise convolution (MixConv) naturally mixes up multiple kernel sizes in a single convolution by splitting up the input of convolution into groups and applying different kernel sizes to each group [32]. ShiftFaceNet is built upon the architectural foundation introduced in ShiftNet for face recognition, leveraging the FLOP-free "shift" operation as a substitute for spatial convolutions [35]. By integrating shifts with point-wise convolutions, an end-to-end trainable shift-based module is developed, which mitigates computational complexity. MobiFace, on the other hand, explores quicker downsampling of feature maps and bottleneck residual blocks to formulate lightweight models [12]. Based on ShuffleNetV2 [20], another lightweight model called ShuffleFaceNet replaces the Global Average Pooling layer with a Global Depth-wise Convolution layer in the original architecture to reduce the parameters [21]. A new family of lightweight architectures tailored for facial recognition, termed PocketNet, was developed using neural architecture search (NAS) [6]. Recently, EdgeFace emerged as a promising model, drawing inspiration from EdgeNext and integrating the strengths of both CNN and Transformer architectures, leading to remarkable performance in face recognition tasks [13]. Other notable lightweight face recognition models, such as GhostFaceNet [2], Proxyless-FaceNAS [7], and VarGFaceNet [36], have also been devised to balance computational complexity with high accuracy benchmarks.

3. Preliminaries and Proposed Method

As mentioned in Section 1, the mainstream approach in ResNet-based face recognition networks involves linearizing the spatial map after the last convolutional layer before passing it to the fully-connected layer. This linearization significantly impacts the model parameters. This significant architectural deviation, introducing the linearization of the spatial map, can be mainly attributed to an inherent flaw in average pooling. Average pooling treats all pixels equally, whereas in face recognition, center pixels are often considered more important than corner pixels. This leads to a suboptimal 512-D feature vector for image representation. To mitigate the limitations of average pooling and take a step towards eliminating the need for linearizing the

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on May 21,2025 at 02:58:53 UTC from IEEE Xplore. Restrictions apply.

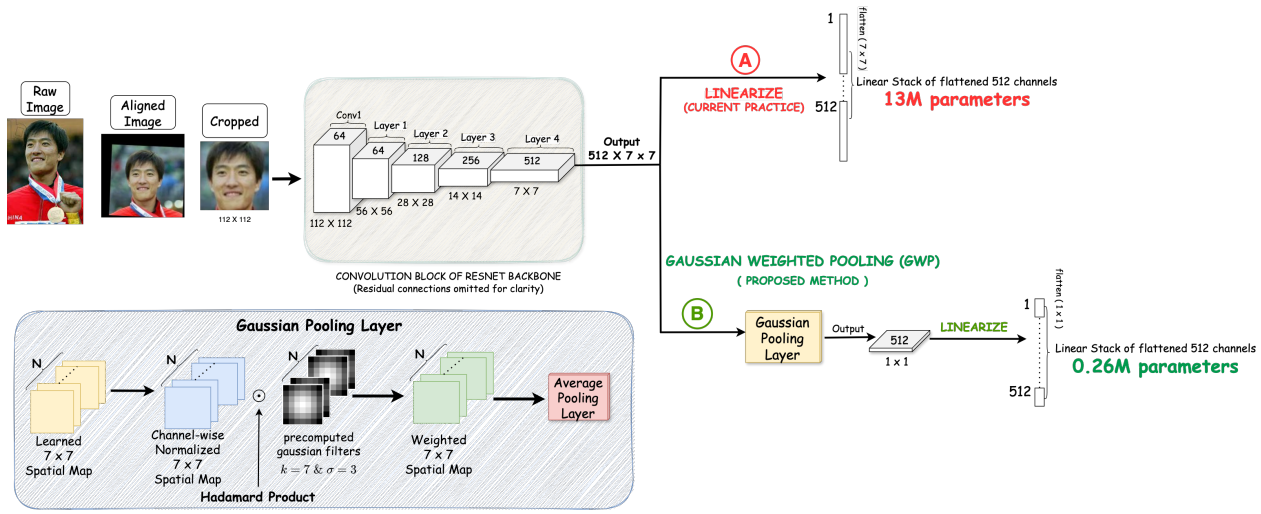


Figure 3. **Linearizing the final output spatial map to obtain a high-quality intermediate 512-D vector, as commonly practiced in standard face recognition methods, results in unneccessary parameters in the linear layer. The utilization of Gaussian Weighted Pooling (GWP) offers a solution to bypass the linearization process while preserving network accuracy with significantly fewer parameters**. Branch Ⓐ, depicted in the figure, illustrates the prevailing standard approach in mainstream ResNet-based face recognition networks. This configuration adds approximately 13 million parameters in the linear layer. Branch Ⓑ introduces the incorporation of the **Gaussian Weighted Pooling (GWP)** layer. This layer downsamples the 7×7 output spatial map to 1×1, which employs a pre-computed Gaussian attention kernel to assign varying importance to pixels based on their position within the map, only adding 0.26 million parameters in the linear layer.

spatial map, we propose a novel method termed **Gaussian Weighted Pooling (GWP)** in this study.

Mathematically, gaussian weighted pooling can be formalized in these following steps. The first step is the feature map normalization step, which can be represented as:

$$\hat{X}_{c,h,w} = \frac{X_{c,h,w} - \mu_c}{\sigma_c} \tag{1}$$

where, $X_{c,h,w}$ is the learned spatial map, $\hat{X}_{c,h,w}$ is the normalized spatial map, μ is mean of the feature map, and σ is the deviation in the feature map. We have used $\sigma = 3$ in our experiments. This sigma value does not excessively smooth or overly sharpen the output spatial map, which has dimensions of 7×7.

Next, the normalized spatial map is passed through a pre-computed Gaussian attention layer. The pre-computed gaussian attention map can be represented as:

$$G_{x,y} = \exp\left(-\frac{(x - x_c)^2 + (y - y_c)^2}{2\sigma^2}\right) \tag{2}$$

where, G denotes the Gaussian attention kernel with a kernel size of k and standard deviation σ. Here, G represents the unnormalized kernel. To maintain the energy of the original signal, ensuring that this attention layer functions as a filter without altering the overall energy of the signal (learned spatial map), we normalize the kernel such that the sum of its elements equals 1. This normalization is illustrated as follows:

$$G_{attention} = \frac{G}{\sum_{x=1}^{n}\sum_{y=1}^{n} G_{x,y}} \tag{3}$$

where, $G_{attention}$ is the normalized Gaussian attention kernel. Finally, the learned attention spatial map is obtained by performing the Hadamard product between the normalized learned spatial map and the Gaussian attention kernel. This is represented as follows:

$$\tilde{X}_{c,h,w} = \hat{X}_{c,h,w} \odot G_{attention} \tag{4}$$

where, $\tilde{X}_{c,h,w}$ is the weighted spatial map, $\hat{X}_{c,h,w}$ is the normalized spatial map and $G_{attention}$ is the normalized Gaussian attention kernel. This weighted spatial map is then subject to Global Average Pooling.

The Figure 3 illustrates the overview of Gaussian Weighted Pooling (GWP). The weighted spatial map ($\tilde{X}_{c,h,w}$) ensures that center pixels are given greater importance, while corner pixels are assigned lower importance. Therefore, performing average pooling on this weighted spatial map is not merely a naïve averaging of all pixels, as each pixel's positional importance is taken into account.

Numerical Stability during Training? This implementation ensures numerical stability during training primarily due to two normalization steps: first, each learned feature map is normalized before the Hadamard product is performed, and second, the Gaussian attention kernel is

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on May 21,2025 at 02:58:53 UTC from IEEE Xplore. Restrictions apply.

Loss	Training	Params	Params. Linear (% total)	LFW	CFP-FP	AGEDB-30	CALFW	CPFLW	Average
ArcFace	Baseline with k = 3	43.6 M	12.8 M (≈ 30%)	99.77 ± 0.03	99.03 ± 0.06	97.56 ± 0.14	95.98 ± 0.08	94.07 ± 0.10	97.28 ± 0.08
	Baseline with k = 7	43.6 M	12.8 M (≈ 30%)	**99.80**	**98.96**	**97.78**	**96.02**	**94.10**	**97.33**
	GWP with k = 3	31 M	0.26 M (≈ 0.84%)	**99.80**	**98.86**	97.23	**95.97**	**94.45**	**97.26**
	GWP with k = 7	31 M	0.26 M (≈ 0.84%)	**99.78**	98.73	**97.50**	**95.82**	**94.67**	**97.30**
AdaFace	Baseline with k = 3	43.6 M	12.8 M (≈ 30%)	99.79 ± 0.02	98.88 ± 0.07	97.64 ± 0.06	96.02 ± 0.10	94.01 ± 0.14	97.26 ± 0.05
	Baseline with k = 7	43.6 M	12.8 M (≈ 30%)	**99.80**	**98.87**	**97.78**	**96.00**	**94.28**	**97.35**
	GWP with k = 3	31 M	0.26 M (≈ 0.84%)	**99.78**	98.69	96.98	**95.85**	**94.55**	**97.18**
	GWP with k = 7	31 M	0.26 M (≈ 0.84%)	**99.82**	**98.73**	97.13	**95.90**	**94.67**	**97.25**

Table 1. Baseline (ResNet50) vs. Altered Backbones on Standard Benchmarks. *Note that most accuracy values for networks with reduced linear layers are all within two standard deviations of the baseline*. Integration of a novel Gaussian Weighted Pooling (GWP) layer, which rectifies the inherent flaws of treating each pixel equally in naïve average pooling, results in a reduction of approximately 29% in the number of parameters within the backbone, all while preserving accuracy. [**Keys**: Gaussian Weighted Pooling (GWP), **lower and within 2 σ of baseline average**, **higher than or equal to baseline average**]

normalized to have a sum of 1. These two normalization steps combined preserve overall energy or magnitude in the resulting spatial map, preventing inconsistencies in the representation of spatial features.

Why GWP is suitable for face recognition? One major difference between face recognition and general object classification lies in the image pre-processing for training and testing. In general object classification, training data is typically randomly cropped on-the-fly from larger images during training. However, in face recognition, both training and test images are first detected, then the face is cropped and aligned to the center of the frame. This process provides a substantial prior information about the positioning of the face. Hence, pre-computed static attention filters, such as those used for Gaussian Weighted Pooling (GWP), are suitable for the domain of face recognition.

4. Implementation Details

In our experiments, we use two kinds of margin loss functions: one with a static margin value - ArcFace, and the other with an adaptive margin value - AdaFace to ensure that the findings are consistent for both learning paradigms. We use ResNet50 [14] with modifications as proposed in [10] as baseline backbone. For the backbone with GWP, the flattening of the final output from convolution layer is replaced with a GWP layer. For ArcFace, we use combined margin values of (1, 0, 0.4). We use cleaned WebFace4M dataset [4, 43] as the training set. Images in WebFace4M dataset are pre-aligned using RetinaFace [9]. The model is trained for 20 epochs using SGD as the optimizer [26], with momentum of 0.9, an initial learning rate of 0.1 and weight decay of 5e-4. We adopt polynomial decay as the learning rate scheduler during training from [1]. For AdaFace, we follow the original paper and use initial margin of 0.4. The model is trained for 26 epochs using SGD as the optimizer, with momentum of 0.9 and initial learning rate of 0.1. The learning rate is reduced by a gamma factor of 0.1 at the 12th, 20th, and 24th epochs.

5. Results

We compare the results for two versions of the backbone: Baseline and Gaussian Weighted Pooling (GWP). We perform the evaluation of baseline model five times to calculate the average performance and its standard deviation. This allows us to determine statistical significance for comparisons. These networks are presented using a kernel size of 3×3 and 7×7 in the first layer, as discussed in Section 3.

Results on Standard Benchmarks. The results for standard benchmarks are in Table 1. First, as hypothesized, it is not very clear whether employing a kernel size of 7×7 in the first convolution layer is better than 3×3 for the baseline model. Nonetheless, the results for both kernel settings are consistently on par with each other. This consistency holds across standard benchmark accuracy and two different loss functions. Second, using Gaussian Weighted Pooling (GWP) reduces the total parameters by 29% for the given configuration of the network (R50). This equivalently means that the number of linear parameters is reduced by 98% (from 13M to 0.26M). For both kernel size of 3×3 and 7×7 with GWP, the average accuracy on standard benchmarks is comparable to the baseline accuracy. However, with GWP, the average accuracy consistently surpasses that of a 3×3 kernel size when employing a 7×7 kernel size. This indicates that employing GWP with a 7×7 kernel can effectively match the performance of a parameter-inflated baseline backbone featuring linearized spatial mapping but with a notable reduction in parameters. This observation remains consistent across both training paradigms: adaptive margin and static margin methods.

Results on IJB Suite. The results for IJB testing are in Table 2. The results closely resemble to what was observed

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on May 21,2025 at 02:58:53 UTC from IEEE Xplore. Restrictions apply.

		IJB-B					IJB-C				
Loss	FAR Threshold →	$1e^{-5}$	$1e^{-4}$	$1e^{-3}$	$1e^{-2}$	$1e^{-1}$	$1e^{-5}$	$1e^{-4}$	$1e^{-3}$	$1e^{-2}$	$1e^{-1}$
ArcFace	Baseline with k = 3	91.54 ± 0.21	95.16 ± 0.09	96.80 ± 0.06	97.83 ± 0.10	98.80±0.02	95.05±0.06	96.88±0.08	97.99±0.04	98.63±0.02	99.28±0.02
	Baseline with k = 7	**91.67**	**95.35**	**96.74**	**97.79**	**98.88**	**94.97**	**96.93**	**97.93**	**98.63**	**99.28**
	GWP with k = 3	90.77	94.70	96.41	**97.94**	**98.93**	94.44	96.47	97.65	**98.61**	**99.30**
	GWP with k = 7	90.32	94.85	96.51	**97.83**	**98.98**	94.31	96.50	97.72	**98.59**	**99.28**
AdaFace	Baseline with k = 3	91.58 ± 0.15	95.41 ± 0.12	97.11 ± 0.02	98.15 ± 0.02	99.03 ± 0.03	95.05 ± 0.04	96.89 ± 0.09	98.08 ± 0.02	98.81 ± 0.06	99.36 ± 0.02
	Baseline with k = 7	**91.93**	**95.38**	**97.14**	**98.22**	**99.05**	**95.08**	**96.91**	**98.09**	**98.81**	**99.39**
	GWP with k = 3	90.50	94.45	96.77	**98.18**	**99.07**	94.07	96.28	97.77	**98.73**	**99.39**
	GWP with k = 7	90.63	94.72	96.83	**98.19**	**99.06**	93.84	96.31	97.80	**98.68**	**99.35**

Table 2. Baseline (ResNet50) vs. Altered Backbones on IJB testing suite. The results align fairly consistently with those obtained from standard benchmark evaluation datasets. Incorporation of a novel Gaussian Weighted Pooling layer, which reduces 29% of the total parameters in the network, shows only marginal effects on accuracy. [**Keys**: Naïve Intermediate Downsample (NID), Gaussian Weighted Pooling(GWP), **lower and within 2 σ of baseline average**, **higher than or equal to baseline average**]

for standard benchmarks for baseline models. Employing a kernel size of 3×3 and 7×7 in the first layer is consistently on par for varying TAR@FAR thresholds. For both kernel settings, the accuracy on IJB is on par with or better than the baseline accuracy at less strict thresholding (0.01 or higher). At stricter thresholds, such as lower than 0.01, the accuracy is lower than the baseline. However, this reduced accuracy can be partially mitigated by using a kernel size of 7×7 in the first convolution layer with GWP. With a kernel size of 7×7 with GWP, the accuracy on varying thresholds, from strict to less strict, is higher than the accuracy with kernel size of 3×3 with GWP and is comparable to the accuracy of the baseline model with a kernel size of 3×3, which is commonly used in face recognition networks today.

6. Ablation Study

Downsampling to 1×1 without GWP.

To underscore the significance of Gaussian Weighted Pooling (GWP), we undertake ablation in two distinct steps. Initially, we replace GWP with GAP in a pre-trained model by eliminating the pre-computed static kernel. This operation remains valid as GWP simply utilizes pre-computed static attention kernels before average pooling, thus there is no alteration in the learnable parameters of the network. Subsequently, we train entire model using naïve average pooling layer (GAP) to downsample the output spatial map from 7×7 to 1×1. The results of this experiment are presented in Table 3. In a pre-trained model trained with GWP, the pre-computed Gaussian kernel holds significant importance. Without this pre-computed static kernel, there is a drastic drop in performance during inference. Results from models trained from scratch with GAP (w/o GWP) also suggest that GWP boosts accuracy by selectively attending to various regions of the output spatial map, prioritizing the center over the corners. Notably, this increase in accuracy is achieved without additional parameters; both networks, whether with GWP or GAP, maintain the same number of learnable parameters.

Loss	Training	Avg.	IJB-B	IJB-C
ArcFace	w GWP	**97.30**	**94.85**	**96.50**
	w/o GWP†	97.02	93.33	95.58
	w GAP‡	97.10	94.25	96.14
AdaFace	w GWP	**97.25**	**94.72**	**96.31**
	w/o GWP†	97.01	94.06	95.76
	w GAP‡	97.11	94.11	95.94

‡ training from scratch with GAP (w/o GWP).

† pre-trained model with GWP without attention kernel during inference.

Table 3. Comparison of model performance with and without Gaussian Weighted Pooling (GWP) layer. GWP significantly improves accuracy without adding parameters and with negligible addition to computation cost. Replacing GWP with naïve average pooling (GAP) in a pre-trained GWP model leads to a significant performance decline. Moreover, models trained using Naïve Average Pooling consistently from scratch also demonstrates inferior performance when compared to those incorporating the GWP layer. For standard benchmarks, 1:1 verification accuracy (%) and TAR@FAR=$1e^{-4}$ are reported respectively. [**Key**: **best**]

7. Comparative Analysis

7.1. Comparison with SoTA Edge Models.

We are not aware of any prior work that has investigated in-depth the archictectural changes made to vanilla ResNet to adapt it for face recognition. While the seminal work by Deng et al. [10] explored various residual structures, it didn't analyze alternatives to linearizing the output convolution spatial map. Therefore, we alter the network's structure to investigate the impact of linearizing the output spatial map. This involves significantly reducing the output spatial map through Gaussian Weighted Pooling (GWP) before linearization. This reduction slashes the

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on May 21,2025 at 02:58:53 UTC from IEEE Xplore. Restrictions apply.

Model	Num. Params. (M)	Size Train. (M)	Benchmark Avg. (%)	IJB-B (%)	IJB-C (%)
ProxylessFaceNAS [22]		5.1	93.01	87.10	89.70
MobileFaceNetV1 [22]		5.1	94.64	92.0	93.90
PocketNetS-256 [6]		5.1	94.75	89.31	91.33
ShuffleFaceNet 1.5x [21, 22]	$\ll$43.6	5.1	95.49	92.30	94.30
PocketNetM-256 [6]		5.1	95.61	90.74	92.70
MobileFaceNet[22]		5.1	95.72	92.80	94.70
VarGFaceNet [22, 36]		5.1	96.04	92.90	94.70
EdgeFace [13]		5.1	96.15	92.67	94.85
ArcFace	43.6	3.9	97.28 ± 0.08	95.16 ± 0.09	96.88 ± 0.08
ArcFace w/ GWP (Ours)	31	3.9	97.30	94.85	96.50
ArcFace w/ GDC	31	3.9	97.32	95.09	96.79
AdaFace	43.6	3.9	97.26 ± 0.05	95.41 ± 0.12	96.89 ± 0.09
AdaFace w/ GWP (Ours)	31	3.9	97.25	94.72	96.31
AdaFace w/ GDC	31	3.9	**96.63**	**93.55**	**95.43**

Table 4. With GWP, the adoption of ResNet for facial recognition becomes more efficient. Our approach, integrating GWP, attains accuracy comparable to that of the heavier model, yet significantly surpasses edge models in accuracy. Additionally, GWP exhibits greater numerical stability and demonstrates consistent performance across both static and adaptive margin methods. For standard benchmarks, 1:1 verification accuracy (%) and TAR@FAR=$1e^{-4}$ are reported respectively.

network's parameters by approximately 12.5 million, and our experiments have shown that the accuracy remains on par with the baseline model. However, this reduction doesn't transform the network into an ultralight model; rather, it crafts an efficient iteration of the deep ResNet model for face. Lightweight models typically employ efficient operations and blocks from the outset, balancing computation complexity with accuracy. To demonstrate the distinct yet efficient nature of our model, we compare its accuracy with other lightweight models. *Our aim isn't to assert superiority over all edge methods but to showcase that our results belong to a distinct model category*. The table clearly demonstrates that our model, as proposed in this study, surpasses all previous lightweight models. Notably, it achieves this while showcasing approximately 29% fewer parameters compared to the adapted ResNet version designed for face recognition, all while maintaining a similar level of performance or having marginal impacts on accuracy.

Comparison with alternative possible approach. In addition to comparing with other edge models, we also train the backbone by substituting Gaussian Weighted Pooling (GWP) with Global Depth Wise Convolution (GDC), which serves as an alternative approach to GWP for reducing linear parameters. Depth-wise separable convolution, initially proposed in MobileNet [15] and later adapted in MobileFaceNet [8], extensively utilizes such convolutions for efficiency. Particularly in MobileFaceNet, the Global Depth Wise Convolution (GDC) layer handles the spatial output from the last convolution layer, similar to the GWP used in our study. However, a key distinction in GDC when compared to GWP is that GDC use learnable spatial kernel. As observed in Table 4, GDC demonstrates comparable performance to the baseline in static margin models such as ArcFace, but experiences catastrophic failure in adaptive margin models such as AdaFace. These findings underscore two key points: a) GWP exhibits numerical stability and yields more consistent outcomes across both adaptive and static margin learning paradigms, and b) although GDC demonstrates instability in adaptive margin models, its utilization in static margin models can offer an additional illustration that linearization of output spatial map might be unnecessary, underscoring the importance of seeking alternative parameter-friendly approaches.

7.2. GWP w/ Deeper and Lighter Backbone.

In this section, we extend our analysis regarding the effectiveness of GWP to more lighter and deeper versions of the SoTA backbone and the results are summarized in Table 5. For a smaller ResNet34 backbone, substituting the linearization with GAP reduces the total number of parameters in the model by about 37% (from 34M to 26M). As a result, there is a reduction in accuracy for both loss functions used. However, with GWP, a significant portion of the lost accuracy is regained. Our experiments shows that 100% of the lost accuracy can be regained for standard benchmarks in the case of AdaFace, along with 52% for IJBB and 50% for IJBC. Similarly, for ArcFace, the accuracy is regained by 100% for standard benchmarks, 22% for IJBB, and 32% for IJBC. Likewise, for a larger ResNet100 backbone, replacing the linearization with Global Average Pooling also

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on May 21,2025 at 02:58:53 UTC from IEEE Xplore. Restrictions apply.

results in a substantial reduction in the total number of parameters in the model by approximately 20% (from 65M to 52M). This change results in a decrease in model accuracy, although it's not as significant as what we observed with the smaller backbone. The larger backbone contains only 20% of its total parameters within the linear layers (13M out of 65M), whereas in the case of the smaller backbone, the linear layers constitute approximately 40% of the total parameters (13M out of 52M). Thus, for the smaller backbone, linear parameters play a pivotal role in learning, whereas for the larger backbone, a significant portion of learning might already be accomplished by convolution parameters. While most of the learning in larger backbones is achieved by the convolution layers, there is still a reduction in performance observed when substituting the linearization of the output spatial map with GAP. However, much of this lost accuracy can be regained by using GWP for larger backbones as well. For larger backbone, 58% of the lost accuracy can be regained for standard benchmarks in the case of AdaFace, along with 66% for IJBB and 64% for IJBC. Similarly, for ArcFace, the accuracy is regained by 95% for standard benchmarks, 68% for IJBB, and 100% for IJBC. Overall, employing GAP instead of linearization leads to a decline in model accuracy, affecting both smaller and larger backbones. Nonetheless, utilizing GWP enables much of the lost accuracy to be recovered without incurring additional parameter costs.

Loss	Network	Training	NP (M)	PL (M) (% total)	B. Avg	IJB-B	IJB-C
ArcFace	ResNet34	Baseline	34.2	12.8 (37.4%)	97.04	94.56	96.29
		w/ GAP	21.6	0.26 (1.2%)	96.89	93.53	95.62
		w/ GWP	21.6	0.26 (1.2%)	97.14	93.75	95.83
	ResNet100	Baseline	65.2	12.8 (19.6%)	97.51	95.15	96.73
		w/ GAP	52.6	0.26 (0.5%)	97.30	94.92	96.61
		w/ GWP	52.6	0.26 (0.5%)	97.50	95.07	96.78
AdaFace	ResNet34	Baseline	34.2	12.8	97.03	94.92	96.49
		w/ GAP	21.6	0.26	96.96	94.21	95.93
		w/ GWP	21.6	0.26	97.04	94.58	96.21
	ResNet100	Baseline	65.2	12.8	97.47	95.74	97.15
		w/ GAP	52.6	0.26	97.35	94.88	96.45
		w/ GWP	52.6	0.26	97.42	95.45	96.90

Table 5. Substituting the linearization of the output spatial map with GAP reduces the model accuracy for both smaller and larger backbones. However, using GWP, the lost accuracy can be regained without requiring additional parameters for both smaller and larger backbones, as well as for both static and adaptive margin learning paradigms. For standard benchmarks, 1:1 verification accuracy (%) and TAR@FAR=$1e^{-4}$ are reported respectively. **Keys**: [NP - Number of Parameters, PL (% total) - Linear Paramaters (% of total parameters)]

8. Conclusions.

Our study reveals that linearizing the output spatial map causes significant parameter inflation in the linear layer of ResNet-based deep networks used for face recognition. To tackle this challenge, we propose a novel Gaussian Weighted Pooling (GWP) layer, which circumvents the necessity for linearization and thereby alleviates parameter inflation. Our ablation studies demonstrate that using naïve average pooling (GAP) reduces the accuracy of the models. However, integrating GWP maintains accuracy across standard benchmarks such as LFW, CFP-FP, AGEDB-30, CALFW, and CPLFW compared to parameter-inflated baseline models, both for static and adaptive margin-based models. Furthermore, even for the more challenging IJB family, there is only a marginal drop in performance, despite a significant reduction of network parameters. Our results underscore that linearizing the output spatial map to attain high-quality vector representation of a facial image is a parameter-unfriendly operation. Other alternatives like the one proposed in this work should be explored to manage parameter count effectively. Additionally, our findings suggest that using GWP with larger 7×7 kernels in the first layer yields notably superior results. One reason for this could be that, with the removal of numerous linear layers - which are known for their tendency to memorize information [3, 37] - the convolution kernel is compelled to learn more effective features. In this context, the larger kernel in the first layer, which interacts directly with raw RGB data, might be capturing superior features due to its larger receptive field.

Future Work. In this study, we have demonstrated the effectiveness of leveraging face recognition-specific image alignment to compute static attention kernels, thereby assigning importance to pixels based on their spatial position in the feature map. In the future, exploring methods such as more advanced and numerically stable kernels, or adaptive normalized kernels, can be investigated to further use the prior information of face alignment to reduce the accuracy gap while keeping the parameter count in check.

References

[1] Insightface: 2d and 3d face analysis project. https://github.com/deepinsight/insightface/. 5

[2] M. Alansari, O. A. Hay, S. Javed, A. Shoufan, Y. Zweiri, and N. Werghi. Ghostfacenets: Lightweight face recognition model from cheap operations. In *IEEE Access*, 2023. 2, 3

[3] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning (ICML)*, pages 233–242, 2017. 8

[4] A. Bhatta, D. Mery, H. Wu, J. Annan, M. C. King, and K. W. Bowyer. Our deep cnn face matchers have developed achromatopsia. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 142–152, 2024. 5

[5] F. Boutros, N. Damer, M. Fang, F. Kirchbuchner, and A. Kui-

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on May 21,2025 at 02:58:53 UTC from IEEE Xplore. Restrictions apply.

jper. Mixfacenets: Extremely efficient face recognition networks. In *International Joint Conference on Biometrics (IJCB)*, pages 1–8, 2021. 2, 3

[6] F. Boutros, P. Siebke, M. Klemt, N. Damer, F. Kirchbuchner, and A. Kuijper. Pocketnet: Extreme lightweight face recognition network using neural architecture search and multistep knowledge distillation. In *IEEE Access*, volume 10, pages 46823–46833, 2022. 3, 7

[7] H. Cai, L. Zhu, and S. Han. Proxylessnas: Direct neural architecture search on target task and hardware. In *International Conference on Learning Representations (ICLR)*, 2019. 3

[8] 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 (CCBR)*, pages 428–438, 2018. 2, 3, 7

[9] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, pages 5203–5212, 2020. 5

[10] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 4690–4699, 2019. 1, 2, 3, 5, 6

[11] Y. Duan, J. Lu, and J. Zhou. Uniformface: Learning deep equidistributed representation for face recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3415–3424, 2019. 1, 2, 3

[12] C. N. Duong, K. G. Quach, I. Jalata, N. Le, and K. Luu. Mobiface: A lightweight deep learning face recognition on mobile devices. In *International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–6, 2019. 3

[13] A. George, C. Ecabert, H. O. Shahreza, K. Kotwal, and S. Marcel. Edgeface: Efficient face recognition model for edge devices. In *Transactions on Biometrics, Behavior, and Identity Science (T-BIOM)*, 2024. 2, 3, 7

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1, 3, 5

[15] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3, 7

[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 07-49, University of Massachusetts, Amherst, October 2007. 1

[17] Y. Huang, Y. Wang, Y. Tai, X. Liu, P. Shen, S. Li, J. Li, and F. Huang. Curricularface: adaptive curriculum learning loss for deep face recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 5901–5910, 2020. 1, 2, 3

[18] M. Kim, A. K. Jain, and X. Liu. Adaface: Quality adaptive margin for face recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 18750–18759, 2022. 1

[19] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 212–220, 2017. 1, 2, 3

[20] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *European Conference on Computer Vision (ECCV)*, pages 116–131, 2018. 3

[21] Y. Martindez-Diaz, L. S. Luevano, H. Mendez-Vazquez, M. Nicolas-Diaz, L. Chang, and M. Gonzalez-Mendoza. Shufflefacenet: A lightweight face architecture for efficient and highly-accurate face recognition. In *International Conference on Computer Vision Workshops (ICCVW)*, pages 0–0, 2019. 2, 3, 7

[22] Y. Martinez-Diaz, M. Nicolas-Diaz, H. Mendez-Vazquez, L. S. Luevano, L. Chang, M. Gonzalez-Mendoza, and L. E. Sucar. Benchmarking lightweight face architectures on specific face recognition scenarios. *Artificial Intelligence Review*, pages 1–44, 2021. 7

[23] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, et al. Iarpa janus benchmark-c: Face dataset and protocol. In *International Joint Conference on Biometrics (IJCB)*, pages 158–165, 2018. 1

[24] Q. Meng, S. Zhao, Z. Huang, and F. Zhou. Magface: A universal representation for face recognition and quality assessment. In *Computer Vision and Pattern Recognition (CVPR)*, pages 14225–14234, 2021. 1, 2, 3

[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 *Computer Vision and Pattern Recognition Workshops (CVPRW)*, volume 2, page 5, 2017. 1

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. 5

[27] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2015. 2

[28] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015. 1, 3

[29] S. Sengupta, J. Cheng, C. Castillo, V. Patel, R. Chellappa, and D. Jacobs. Frontal to profile face verification in the wild. In *Winter Conference on Applications of Computer Vision (WACV)*, February 2016. 1

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 2

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. 2

[32] M. Tan and Q. V. Le. Mixconv: Mixed depthwise convolutional kernels. In *British Machine Vision Conference (BMVC)*, 2019. 3

[33] P. Terhörst, M. Ihlefeld, M. Huber, N. Damer, F. Kirchbuchner, K. Raja, and A. Kuijper. Qmagface: Simple and accu-

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on May 21,2025 at 02:58:53 UTC from IEEE Xplore. Restrictions apply.

rate quality-aware face recognition. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 3484–3494, 2023. 1, 2, 3

[34] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen, et al. Iarpa janus benchmark-b face dataset. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 90–98, 2017. 1

[35] B. Wu, A. Wan, X. Yue, P. Jin, S. Zhao, N. Golmant, A. Gholaminejad, J. Gonzalez, and K. Keutzer. Shift: A zero flop, zero parameter alternative to spatial convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 9127–9135, 2018. 2, 3

[36] 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 *International Conference on Computer Vision Workshops (ICCVW)*, pages 0–0, 2019. 2, 3, 7

[37] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. 8

[38] P. Zhang, F. Zhao, P. Liu, and M. Li. Efficient lightweight attention network for face recognition. *IEEE Access*, 10:31740–31750, 2022. 2, 3

[39] K. Zhao, J. Xu, and M.-M. Cheng. Regularface: Deep face recognition via exclusive regularization. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1136–1144, 2019. 1, 2, 3

[40] T. Zheng and W. Deng. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. Technical Report 18-01, Beijing University of Posts and Telecommunications, February 2018. 1

[41] T. Zheng, W. Deng, and J. Hu. Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv preprint arXiv:1708.08197*, 2017. 1

[42] J. Zhou, X. Jia, Q. Li, L. Shen, and J. Duan. Uniface: Unified cross-entropy loss for deep face recognition. In *International Conference on Computer Vision (ICCV)*, pages 20730–20739, 2023. 1, 2, 3

[43] Z. Zhu, G. Huang, J. Deng, Y. Ye, J. Huang, X. Chen, J. Zhu, T. Yang, J. Lu, D. Du, et al. Webface260m: A benchmark unveiling the power of million-scale deep face recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 10492–10502, 2021. 5

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on May 21,2025 at 02:58:53 UTC from IEEE Xplore. Restrictions apply.