SwiftFaceFormer: An Efficient and Lightweight Hybrid Architecture for Accurate Face Recognition Applications

Luis S. Luevano[1][0000−0001−5784−0826], Yoanna Martínez-Díaz[2][0000−0002−1591−5989], Heydi Méndez-Vázquez[2][0000−0002−7834−1791], Miguel González-Mendoza[3][0000−0001−6451−9109], and Davide Frey[1][0000−0002−6730−5744]

[1] Univ Rennes, Inria, CNRS, IRISA, France
{luis-santiago.luevano-garcia, davide.frey}@inria.fr
[2] Advanced Technologies Application Center (CENATAV) 7A #21406 Siboney, Playa, P.C.12200, Havana, Cuba
{ymartinez, hmendez}@cenatav.co.cu
[3] Tecnologico de Monterrey, School of Engineering and Sciences, Monterrey, Nuevo León, 64849, Mexico
mgonza@tec.mx

Abstract. With the growing breakthrough of deep learning-based face recognition, the development of lightweight models that achieve high accuracy while maintaining computational and memory efficiency has become paramount, especially for deployment on embedded domains. While Vision Transformers have shown significant promise results in various computer vision tasks, their adaptability to resource-constrained devices remains a significant challenge. This paper introduces SwiftFaceFormer, a new efficient, and lightweight family of face recognition models inspired by the hybrid SwiftFormer architecture. Our proposal not only retains the representational capacity of its predecessor but also introduces efficiency improvements, enabling enhanced face recognition performance at a fraction of the computational cost. We also propose to enhance the verification performance of our original most lightweight variant by using a training paradigm based on Knowledge Distillation. Through extensive experiments on several face benchmarks, the presented SwiftFaceFormer demonstrates high levels of accuracy compared to the original SwiftFormer model, and very competitive results to state-of-the-art deep face recognition models, providing a suitable solution for real-time, on-device face recognition applications.

Keywords: Lightweight Face Recognition, Efficient Vision Transformer, Knowledge Distillation, Efficient Face Transformer

1 Introduction

In the last decade, deep learning methods based on Convolutional Neural Networks (CNNs) have revolutionized the face recognition research landscape, achieving impressive levels of accuracy compared to "shallow" methods [29]. However,

this increased performance often relies on a high model complexity, which makes it difficult to deploy on embedded devices or smartphones with memory and computational constraints, resulting in finding a suitable trade-off between speed and accuracy to be a significant challenge.

Designing efficient face recognition solutions, from lightweight deep learning architectures proposed for common computer vision tasks, has emerged as a great promising option. Models such as MobileFaceNet [4], ShuffleFaceNet [19], VarGFaceNet [31], MixFaceNets [2] and GhostFaceNets [1] have been built from MobileNetV2 [23], ShuffleNetV2 [17], VarGNet [32], MixNets [27] and GhostNets [11, 28], respectively, reaching high levels of recognition accuracy with a low number of parameters and computational complexity.

On the other hand, it has been recently demonstrated that transformer-based architectures can be incorporated into face recognition with promising results [35]. Although these methods are capable of capturing long-range relations among facial regions, the associated high computational costs due to the effective use of self-attention computation have restricted their usage in resource-limited domains. To address this issue, new hybrid models [9, 12, 15, 26], that combine the strengths of both lightweight CNNs and Vision Transformer (ViT), have been introduced for face recognition, demonstrating that it is possible to meet real-time deployment in practical applications.

Recently, the SwiftFormer network [25] introduced a consistent hybrid design with an efficient additive attention mechanism to model the contextual information with linear complexity. Experiments on image classification, object detection, and segmentation tasks showed that this model achieves state-of-the-art (SOTA) performance, obtaining a good trade-off between accuracy and latency.

In this work, we present a new family of efficient and lightweight hybrid face models, namely SwiftFaceFormer, including five model variants with different levels of complexity. We adopt SwiftFormer [25] as a baseline network structure and adapt it for face recognition applications. Specifically, we leverage a Global Depthwise Convolution (GDC) layer followed by a convolution layer of size 1×1 and a batch normalization layer to produce a compact 512-dimensional feature vector in the embedding process. In addition to the four variants of the SwiftFormer model (XS, S, L1, L3), we introduce a new model variant (XXS), with lower computational complexity in terms of the number of floating-point operations (FLOPs), number of parameters, and model size. To enhance the recognition performance of this compact model, we apply hard knowledge distillation (KD) [3] to train our SwiftFaceFormer-XXS model to learn similar feature representations to the ones learned by a high-performance heavy network. Experiments on challenging benchmarks demonstrate the effectiveness and efficiency of SwiftFaceFormer in comparison to SOTA lightweight CNNs, vision transformers, and hybrid models, showing its potential for deployment on resource-constrained face recognition applications.

The main contributions of our work are summarized as follows:

- We introduce a novel lightweight hybrid face architecture, called SwiftFaceFormer, which extends the efficient SwiftFormer network to the specific do-

main of face recognition for real-time applications. The proposed hybrid network architecture leverages CNN and ViT capabilities through five model variants of different complexities.
- We extend the SwiftFaceFormer family of networks including an extremely-lightweight variant, named SwiftFaceFormer-XXS, that introduces an Efficient Convolutional Encoder with variable convolutional groups per stage. This approach heavily improves efficiency over the original SwiftFormer Convolutional Encoder and the rest of the SwiftFormer variants.
- To enhance the interpretation ability and the recognition performance of the most compact variant of our SwiftFaceFormer model (XXS), we apply the knowledge distillation paradigm. We provide two ablation studies about the effect of using different teacher models to learn feature representations and two different loss functions.
- We provide extensive experiments and comparisons with SOTA face models on different datasets including large-scale face recognition benchmarks such as IJB-B and IJB-C, showing the advantages of our proposed models in terms of both accuracy and efficiency.

The paper is organized as follows. Section 2 reviews the existing lightweight CNNs and ViT models for face recognition. Section 3 introduces the lightweight hybrid SwiftFaceFormer models tailored for face recognition. Experiments are presented in Section 4, followed by discussion and conclusion in Section 5.

2 Related work

In this section, we summarize existing approaches for developing face recognition models with low computational complexity that can be deployed on resource-restricted domains such as embedded devices or smartphones. We also give an overview of ViT models that have been proposed for face recognition, including those based on lightweight face recognition models.

2.1 Lightweight CNNs for face recognition

Designing small and efficient network architectures that reduce the computational effort in comparison to larger and more complex CNN models, has become a promising solution in recent years to achieve a better balance between speed and accuracy. In particular, for face recognition, the most common approach has been modifying lightweight networks originally designed for common computer vision tasks to the specific case of face recognition.

MobileFaceNet [4] and ShuffleFaceNet [19], which are based on MobileNetV2 [23] and ShuffleNetV2 [17], respectively, replace the Global Average Pooling (GAP) layer for a Global Depth-wise Convolution layer, and use the Parametric Rectified Linear Unit (PReLU) activation function instead of the Rectified Linear Unit (ReLU) function. Moreover, they adopt a fast downsampling strategy at the beginning of the networks, an early dimension-reduction strategy at the

last several convolutional layers, and a linear 1×1 convolution layer following a linear GDC layer as the feature output layer.

VarGFaceNet [31] improves the discriminative ability of VarGNet [32] by using an efficient variable group convolutional network for lightweight face recognition. In addition, to improve the interpretation ability of this lightweight network, a recursive knowledge distillation strategy is introduced. In a similar way, MixFaceNets [2] extend the MixConv [27] block with a channel shuffle operation aiming at increasing the discriminative ability. More recently, GhostFaceNets [1] extends two efficient neural architectures, GhostNetV1 [11] and GhostNetV2 [28], by replacing the GAP layer and the pointwise convolution layer with a modified GDC layer. They employ the PReLU activation function and replace the fully connected layers in the squeeze and excitation (SE) modules by convolution layers, to improve the discriminative power of their method.

Another strategy has been using Neural Architecture Search (NAS) [8] to automatically create efficient artificial neural networks specifically designed for face recognition. A family of extremely lightweight face models, namely PocketNets, was proposed in [3], aiming at automating the process of designing a neural network that works effectively. The authors also introduce a novel training paradigm based on knowledge distillation to ease the challenges caused by the significant gap between the teacher and student models, reducing the trade-off between model performance and compactness.

2.2 Vision Transformers for face recognition

In recent years, there has been a growing interest in the use of Vision Transformers (ViT) for different computer vision tasks, including face recognition. FaceTransformer [35] was the first attempt to investigate the performance of ViT models in face recognition, by introducing a Transformer model that uses sliding patches to capture inter-patch information from faces. Although this method achieves comparable performance to state-of-the-art CNNs, it is computationally heavy and unsuitable for low-resource environments.

Recently, CFormerFaceNet [12] combines a lightweight CNN face model with ViT. The authors designed a Group Depth-Wise Transpose Attention that used the CNN's ability to extract local facial features and the Transformer's capability to model global facial features, with lightweight modifications reducing computation requirements. In MobileFaceFormer [15], another hybrid method, both CNN and Transformer branches are parallelized in a dual branch design, and a bi-directional feature fusion bridge connecting dual branches is designed to concurrently retain local facial features and global facial interpretations. A convolutional token initialization method is proposed at the Transformer branch to perceive long-range facial information, enhancing feature interpretations. The CNN branch uses Depth-Wise Separable convolution and attention mechanisms are adopted to improve local facial feature extraction before an Attentive Global Depthwise Convolution (AGDC).

EdgeFace [9] presents a new hybrid model that adapts the EdgeNeXt architecture [18] for face recognition and introduces a Low Rank Linear (LoRaLin)

module to further reduce the computation in linear layers while providing a minimal compromise to the performance of the network. In addition, a split depth-wise transpose attention (STDA) encoder is proposed to process input tensors and encode multi-scale facial features, while maintaining low computational costs and compact storage requirements.

HOTformer [26] is another novel face recognition model based on MobileFaceNets and ViTs that can effectively generate discriminative face representations by regional interaction of faces. The authors introduce two cooperation types of tokens named atomic tokens and holistic tokens to capture the region relationship of the face. Specifically, atomic tokens are generated by fixed-size patches to carry the fine-grained core representation, while holistic tokens are generated from adaptively spatial regions to aggregate information from several facial parts.

3 Approach

In this section, we detail the SwiftFaceFormer architecture specifically tailored for face recognition tasks. This approach is directly inspired by the SwiftFormer network [25], which achieves state-of-the-art performance in general-purpose computer vision tasks. Thus, we first describe the original SwiftFormer architecture, followed by the modifications introduced to make it an efficient and accurate face recognition model.

3.1 SwiftFormer network

The SwiftFormer architecture [25] is a lightweight hybrid design that combines the strengths of CNNs and transformers for real-time mobile vision applications. It builds on EfficientFormer [16] and improves the token mixing by using a simple yet effective Convolutional Encoder. This encoder replaces 3×3 average pooling layers used as a local token mixer by depth-wise convolutions, without increasing the parameters and latency. Moreover, SwiftFormer introduces an efficient additive attention module in the SwiftFormer Encoder module, to model the contextual information with linear complexity, that can be incorporated in all stages of the network. This leads to more consistent learning of local-global representations and significantly reduces the computational complexity.

The original SwiftFormer authors proposed four different configurations for this architecture, varying in complexity, named L3, L1, S, and XS. All of these versions use the same operators although with different depth and width levels. The depth configuration regulates the number of encoding operations, while the width level dictates the number of channels present in the feature map. In each stage, the network performs the Convolutional Encoder operations followed by the SwiftFormer Encoder step. After three stages of stacking multiple Encoder blocks, depending on the complexity, the feature map is averaged and its output is sent to a Linear layer for classification tasks. This output before the linear layer is a Global Average Pooling (GAP) operation, common in other general-purpose architectures.

3.2 SwiftFaceFormer architecture

In order to adapt the SwiftFormer architecture to the face recognition task, we introduce specific refinements to the original approach. As noted in previous works, the GAP operator is less effective for the face recognition task. The reason for this is that the averaging operation weights all the inputs from the feature map equally, which hinders the projection of non-linear features and the capacity to extract discriminative information present in face recognition scenarios in the wild. This shortcoming is accentuated when using the GAP layer output before a Linear layer for final classification purposes. To mitigate this limitation, an alternative is to use an embedding head including a Batch Normalization and a Dropout step before the Linear layer [9].

Recent approaches [4, 19, 31] adopt the Global DepthWise Convolution to spatially reduce the feature map size and adjust the embedding dimension to the final embedding. To extend the SwiftFormer model to the face recognition scenario and improve its performance, we opt to adjust the output channels of the final SwiftFormer Encoding stage to the pre-defined face embedding dimension C using an efficient 1×1 point-wise convolution. We then employ the GDC layer with a 4×4 kernel size, reducing the spatial dimension from the SwiftFormer feature map to a vector, and employ another efficient 1×1 point-wise convolutional operator to output the final face embedding.

3.3 SwiftFaceFormer-XXS

When assessing SwiftFormer's efficiency performance, we noted a heavy load of convolutional operations in the Convolutional Encoders from the SwiftFormer architecture. As such, we analyzed the computation load in each one of the stages, noting that the earliest stages performed the most computations due to the larger spatial feature map sizes with two point-wise convolutional layers. Our approach consists of converting the last point-wise layer of the Convolutional Encoder into a grouped point-wise convolution. Our intuition for selecting the last point-wise convolutional layer for grouped convolutions instead of the first one, was to retain more input information with full convolutional operations and reduce the compromise on accuracy when using the following grouped convolution. We corroborated the effectiveness of our selection through experimentation.

We propose to employ a descending strategy for the number of groups at each stage. Using larger group sizes for the first stages heavily reduces the computation load and parameter count and leaving deeper stages less compromised achieves a reasonable balance between efficiency and accuracy. The last stage is left uncompromised with regular point-wise convolutions (groups $g = 1$) before the output to our face embedding head. Figure 1 shows our approach to this efficient Convolutional Encoder.

In addition, we reduced the depth regulating the number of encoding operations in stages 2 to 4, maintaining most of the operations in the third stage as in the original architecture. This further reduces the computation load with limited compromises to accuracy. Lastly, we adjusted the width (channels) of the feature

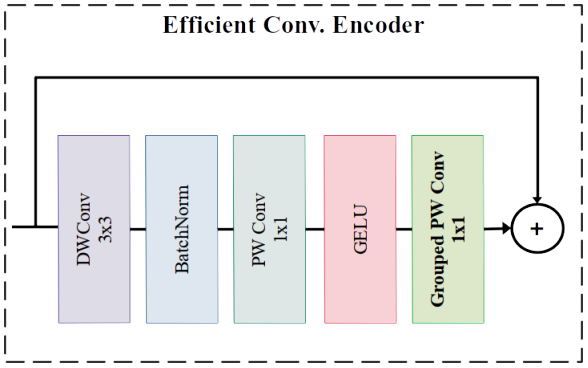

Fig. 1. Our Efficient Conv. Encoder in SwiftFaceFormer-XXS. We employ Grouped Point-Wise Convolutions only at the last layer for maximizing efficiency and mitigating accuracy penalties.

maps starting from 16 in stage one to 128 in the last stage. We selected 128 as our final channel dimension as previous work [19] has suggested that 128 suffices for efficiently embedding facial features. Figure 2 illustrates the modifications for this efficient approach.

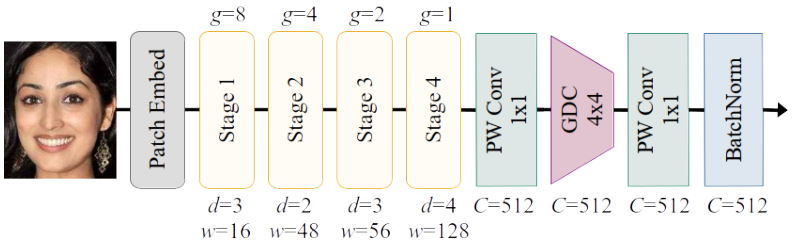

Fig. 2. SwiftFaceFormer-XXS overall architecture. Consistent with the original SwiftFormer notation for the stages, the complexity is expressed as depth d for the number of encoding operations and width w for the number of feature map channels. C denotes the embedding channel dimension for our face recognition head.

4 Experiments

In this section, we introduce the experimental setup of our proposed SwiftFaceFormer models and evaluate their recognition performance over several face benchmark datasets. In addition, we validate the accuracy improvements of the designed SwiftFaceFormer-XXS architecture through two ablation studies.

4.1 Datasets

We used the MS1M-RetinaFace dataset (MS1MV3) [6, 7] for fine-tuning our SwiftFaceFormer models (XXS, XS, S, L1, L3) to the face recognition task. We choose to use this particular dataset to allow a fair evaluation of our method with the rest of the state-of-the-art face recognition benchmarks [7, 14]. This dataset is a clean version of the MS-Celeb-1M dataset [10], which contains 5.1 million of face images collected from 93,431 identities.

To evaluate the effectiveness and robustness of trained SwiftFaceFormer models, we employed several benchmarks including Labeled Faces in the Wild (LFW) [13], Celebrities in Frontal-Profile in the Wild (CFP-FP) [24], AgeDB-30 [22], Cross-age LFW (CALFW) [34], Cross-Pose LFW (CPLFW) [33], IARPA Janus Benchmark-B (IJB-B) [30] and IARPA Janus Benchmark-C (IJB-C) [21].

4.2 Implementation details

For training our approach, we adopt a Stochastic Gradient Descent (SGD) optimizer with a batch size of 3×128 to improve training stability. We perform training on three Nvidia GeForce GTX A6000 GPUs. The learning rate is initialized to 0.05 and decreased by a factor of 10 periodically at epochs 8, 20, 25, and 30. The momentum parameter is set to 0.9 and weight decay at 5e-4. The parameter-initialization method for convolutions is Xavier with random sampling from a Gaussian normal distribution. We use the ArcFace [6] loss function with an angular margin m = 0.5, which turned out to be the best for face recognition. All experiments are implemented on the Pytorch framework. We adopted the pre-trained weights on ImageNet from the original SwiftFormer models [25] to initialize our networks to achieve the best performance on face recognition tasks. During inference, the classification head of the SwiftFaceFormer models is removed and the resulting 512-D embedding is used for the comparisons.

On our Knowledge-Distillation approach for SwiftFormer-XXS, we employed hard-sample distillation with two separate headers, as in [25]. For optimizing the embeddings, we used the Mean Squared Error Loss scaled to 10^4 and the Cosine Distance scaled to 64, when applicable. A second separate header is added to compute the ArcFace loss. Each loss value is scaled to 0.5 for our optimization process. For verification, the embedding from both headers is averaged and used for similarity scoring.

All face images used for both training and testing are detected and aligned by RetinaFace [5]. We use five facial landmarks predicted from RetinaFace to generate the face crops of 112×112, where each pixel (in [0, 255]) is normalized by subtracting 127.5 and then dividing by 128.

4.3 Results

We now present and discuss our experimental results focusing on comparisons with the original general-purpose SwiftFormer architecture and comparing our

approach with state-of-the-art face-recognition models. We also conduct an ablation study of our Knowledge Distillation approach for bridging the accuracy gap between SwiftFormer-XXS and more computationally expensive models.

Comparison with SwiftFormer architecture. To show the advantages of our proposed SwiftFaceFormer architecture for the specific case of face recognition, we compare it with the original SwiftFormer network. For a fair comparison, we trained SwiftFormer models (XS, S, L1, L3) under the same training setting as our SwiftFaceFormer models. In Table 1, we show the verification accuracy of the models on LFW, CFP-FP, AgeDB-30, CALFW and CPLFW datasets. In addition, the number of parameters (Params.) and the MFLOPs are given. It can be seen from the table that, for all variants (XS, S, L1, L3), the proposed SwiftFaceFormer outperforms the original SwiftFormer models, maintaining a very similar computational complexity.

To validate our intuition for selecting the second point-wise convolutional layer as a grouped convolution instead of the first layer, in our Efficient Convolutional Encoder, we trained the SwiftFormer-XXS approach on MS1MV3 and tested face verification on the same datasets as above. Testing both possibilities, we found our approach yields an average increase of 0.5%, with a more notable verification accuracy difference of 1.55% and 1.27% on CPLFW and CFP-FP, respectively.

Method	LFW	CFP-FP	AgeDB-30	CALFW	CPLFW
SwiftFormer-L3	99.6	96.3	96.6	95.7	89.4
SwiftFaceFormer-L3 (ours)	**99.8**	**97.8**	**97.6**	**96.0**	**90.7**
SwiftFormer-L1	99.5	95.4	95.9	95.6	88.9
SwiftFaceFormer-L1 (ours)	**99.7**	**96.7**	**97.0**	**95.8**	**90.1**
SwiftFormer-S	99.4	94.3	94.6	94.9	87.7
SwiftFaceFormer-S (ours)	**99.6**	**95.4**	**95.9**	**95.3**	**88.7**
SwiftFormer-XS	99.4	95.1	94.9	95.0	88.6
SwiftFaceFormer-XS (ours)	**99.6**	**95.5**	**96.4**	**95.4**	**88.7**

Table 1. Comparison of the proposed SwiftFaceFormer with the original SwiftFormer models on popular face recognition benchmarks.

Comparison with the state-of-the-art. Table 2 presents a comparison between our proposed SwiftFaceFormer models (XXS, XS, S, L1, and L3) and previous state-of-the-art CNNs, transformer-based, and hybrid models on seven face-recognition benchmarks. The existing models are ordered according to the number of parameters (compactness), showing those with less than 4M parameters. Our most efficient SwiftFaceFormer models are presented at the end of the

table. SwiftFaceFormer-XS and SwiftFaceFormer-XXS-KD are in with less than 4M. In the case of HOTformer-Net models [26], the number of parameters is unknown, however, the authors used HOTformer-Net (base) and HOTformer-Net (small) for comparisons with state-of-the-art lightweight models.

Verification results from this table reveal that our SwiftFaceFormer models obtain comparable performance to SOTA face recognition models from the literature. Among our models belonging to the first category, SwiftFaceFormer-L3 achieves the best performance. Although it is the most complex of our models, SwiftFaceFormer-L3 (28M parameters) achieves comparable results to other deeper CNN and ViTs models with more than twice number of parameters and 10 times more FLOPs. For example, SwiftFaceFormer-L3 outperformed T2T-ViT, ViT-P10S8, and ViT-P8S8 models on the challenging CFP-FP and CALFW

Method	Type	Params. (M)	MFLOPs	LFW	CFP-FP	AgeDB-30	CALFW	CPLFW	IJB-B	IJB-C
ResNet100-ElasticFace [14]	CNN	65.2	24211.8	99.8	98.7	98.3	96.2	93.2	95.4	96.7
ResNet100-ArcFace [14]	CNN	65.2	24211.8	99.8	98.3	98.2	95.5	92.1	94.2	95.6
T2T-ViT [35]	ViT	63.5	25400	99.8	96.6	98.1	95.8	93.0	-	95.7
ViT-P10S8 [35]	ViT	63.3	24800	99.8	96.4	97.8	95.9	92.9	-	96.1
ViT-P12S8 [35]	ViT	63.3	24800	99.8	96.8	98.1	96.2	93.1	-	96.3
ViT-P8S8 [35]	ViT	63.2	24800	99.8	96.2	97.8	95.9	92.5	-	96.0
ResNet50-Q8-bit [14]	CNN	43.6	-	99.8	97.7	98.0	96.0	92.2	94.3	95.7
ResNet18-Q8-bit [14]	CNN	24.0	1810	99.6	94.5	97.0	95.7	89.5	91.6	93.6
GhostFaceNetV2-1 [1]	CNN	6.9	272.1	99.9	99.3	98.6	96.1	94.7	96.5	97.8
GhostFaceNetV2-2 [1]	CNN	6.8	76.5	99.7	99.3	96.8	95.7	90.2	91.9	93.2
VarGFaceNet [31, 14]	CNN	5.0	1022	99.8	98.5	98.2	95.2	88.6	92.9	94.7
GhostFaceNetV1-1 [1, 14]	CNN	4.1	215.7	99.7	96.8	98.0	95.9	91.9	93.1	94.9
GhostFaceNetV1-2 [1, 14]	CNN	4.1	60.3	99.7	93.3	96.9	95.6	90.1	91.3	93.5
HOTformer-Net (large) [26]	Hybrid	-	2840	99.8	98.8	98.2	95.9	92.9	95.3	96.6
MixFaceNet-M [2]	CNN	3.9	626.1	99.7	-	97.1	-	-	91.6	93.4
EdgeFace-S [9]	Hybrid	3.7	306.1	99.8	95.8	96.9	95.7	92.6	93.6	95.6
MixFaceNet-S [2]	CNN	3.1	451.7	99.6	-	96.6	-	-	90.2	92.3
ShuffleFaceNet [19, 20]	CNN	2.6	577.5	99.7	96.3	97.3	95.1	88.5	92.3	94.3
MobileFaceNet [4, 20]	CNN	2.0	933.3	99.7	96.9	97.6	95.2	89.2	92.8	94.7
EdgeFace-XS [9]	Hybrid	1.8	154	99.7	94.4	96.0	95.3	91.8	92.7	94.9
CFormerFaceNet [12]	Hybrid	1.7	40.0	99.7	95.1	97.1	95.8	90.2	-	-
PocketNetM-128-KD [3]	CNN	1.7	1099	99.7	95.1	96.8	95.7	90.0	90.6	92.6
MobileFaceFormer [15]	Hybrid	1.4	-	99.6	96.8	97.7	96.0	98.4	-	-
MixFaceNet-XS [2]	CNN	1.0	161.9	99.6	-	95.8	-	-	88.5	90.7
PocketNetS-128 [3]	CNN	0.9	587.1	99.5	93.8	95.9	95.0	88.9	88.3	90.8
PocketNetS-128-KD [3]	CNN	0.9	587.1	99.6	94.2	96.1	95.5	89.6	89.4	91.6
HOTformer-Net (base) [26]	Hybrid	-	1301	99.7	97.8	97.6	96.0	91.9	93.8	95.5
HOTformer-Net (small) [26]	Hybrid	-	765	99.7	96.5	96.9	95.6	91.1	92.5	94.5
SwiftFaceFormer-L3 (ours)	Hybrid	28.0	2,015.6	99.8	97.8	97.6	96.0	90.7	92.9	94.7
SwiftFaceFormer-L1 (ours)	Hybrid	11.8	804.6	99.7	96.7	97.0	95.8	90.1	91.8	93.8
SwiftFaceFormer-S (ours)	Hybrid	6.0	485.2	99.6	96.5	96.8	95.8	90.0	91.6	93.5
SwiftFaceFormer-XS (ours)	Hybrid	3.4	293.7	99.6	95.5	96.4	95.4	88.7	90.2	92.3
SwiftFaceFormer-XXS-KD (ours)	Hybrid	1.5	64.1	99.4	92.5	94.8	94.8	87.0	87.8	90.3

Table 2. Comparison with the state-of-the-art CNN, ViT, and hybrid models on popular face recognition benchmarks. The models are ordered based on the number of parameters and divided into $>$ 4M and $<$ 4M parameters. IJB-B and IJB-C correspond to the verification TAR at FAR=1e-4 on the IJB-B and IJB-C datasets

datasets. Our other two models, SwiftFaceFormer-L1 (11.8M parameters) and SwiftFaceFormer-S (6M parameters) perform very similarly to ResNet18-Q8-bit (24M parameters), obtaining even higher verification scores under pose variations from CFP-FP and CPLFW. Moreover, the SwiftFaceFormer-S model reaches the accuracy levels of GhostFaceNetV2-2.

For the second category, the performance of our SwiftFaceFormer-XS and SwiftFaceFormer-XXS models, demonstrate promising results on the evaluated benchmarks. For instance, SwiftFaceFormer-XS obtains as good verification results as the hybrid EdgeFace-S model and the lightweight MixFaceNet-S CNN model. Also, it is able to achieve competitive results with respect to ResNet18-Q8-bit, which belongs to the first category. The use of the KD paradigm allows us to enhance the performance of our compact SwiftFaceFormer-XXS model, offering a good trade-off between efficiency and accuracy for deploying it in limited-resource devices.

As it can be appreciated, in general, we have developed novel hybrid face recognition models that perform well compared to the state-of-the-art, which demonstrates that combining the strengths of both lightweight CNNs and transformers makes it possible to reduce the computational requirements for practical applications.

4.4 Ablation study

The knowledge distillation (KD) paradigm enables a student model to learn from a teacher model, making it a popular technique for training lightweight models from more complex ones. Intending to enhance the performance of our compact model, SwiftFaceFormer-XXS, this section presents two ablation studies based on the hard simple distillation method. First, we show the effect of using different teacher networks for transferring their interpretation capabilities. Then, we evaluate the impact of using different loss functions during KD training.

Method	LFW	CFP-FP	AgeDB-30	CALFW	CPLFW	IJB-B	IJB-C
SwiftFaceFormer-XXS (no KD)	99.2	90.9	92.8	94.0	85.7	81.1	82.8
ResNet100-ArcFace (Teacher)	99.8	98.3	98.2	95.5	92.1	94.2	95.6
SwiftFaceFormer-XXS (student)	**99.4**	92.0	**94.9**	**94.8**	86.4	87.3	89.8
SwiftFaceFormer-L3 (Teacher)	99.8	97.8	97.6	96.0	90.7	92.9	94.7
SwiftFaceFormer-XXS (student)	**99.4**	**92.5**	94.8	**94.8**	**87.0**	**87.8**	**90.3**

Table 3. Results obtained by using KD for training SwiftFaceFormer-XXS model with different teacher models (R100-ArcFace and SwiftFaceFormer-L3) on popular face recognition benchmarks.

Using different teacher networks. We conduct experiments to investigate the effect of different teacher models on SwiftFaceFormer-XXS. We employ two

pretrained and fully converged teacher networks, ResNet100-ArcFace [6] and our largest model SwiftFaceFormer-L3, respectively, to measure the relationship between teachers and student structures. Table 3 presents the recognition results on several face datasets. It can be observed that introducing KD into the SwiftFaceFormer-XXS training phase improves the achieved verification performance of SwiftFaceFormer-XXS (no KD) on all evaluation benchmarks, especially on the large-scale IJB-B and IJB-C databases. Although the performance of the teacher models is very similar, when SwiftFaceFormer-XXS is trained with KD using SwiftFaceFormer-L3 as the teacher network, the verification results are higher. This shows that using a simplified version of the teacher network as student, reduces the model capacity gap between a large deep neural network and a small student neural network.

Using different loss functions. To investigate the effect of loss functions, we train SwiftFaceFormer-XXS models with KD using Mean Square Error (MSE) and Cosine (COS) loss functions, respectively. Comparing the obtained results in Table 4, we can appreciate that for both loss functions, SwiftFaceFormer-XXS achieves very similar results. We choose the SwiftFaceFormer-XXS model trained with the MSE loss function since it offers more stable results during training.

Method	LFW	CFP-FP	AgeDB-30	CALFW	CPLFW	IJB-B	IJB-C
SwiftFaceFormer-L3 (Teacher)	99.8	97.8	97.6	96.0	90.7	92.9	94.7
SwiftFaceFormer-XXS(student)-MSE	99.4	**92.5**	94.8	**94.8**	**87.0**	**87.8**	**90.3**
SwiftFaceFormer-XXS(student)-COS	**99.5**	92.3	**95.2**	**94.8**	86.6	**87.8**	**90.3**

Table 4. Comparison of the results obtained by using different loss functions (MSE and COS) for KD training of SwiftFaceFormer-XXS model on popular face recognition benchmarks.

Efficiency assessment. To support our claim on real-time performance on edge device hardware, we performed latency experiments on our proposal. Table 5 presents the latency and single image throughput (FPS) for our SwiftFaceFormer models on the Nvidia Jetson Nano edge device. We note that our SwiftFaceFormer-XXS-KD exhibits the lowest latency and the highest FPS. We also included the Average FR accuracy of the benchmarks of our method from Table 2 and divided it for the inference latency of our methods, calculating an "Accuracy per latency" score, to better assess the performance gains of our proposed method. We note a huge improvement of Accuracy per latency points with the XXS-KD variant, demonstrating its feasibility for usage on real-time hardware-constrained deployments.

Method	Latency (ms)	FPS throughput	Params (M)	FLOPs (M)	Avg. FR Acc.	Acc. per latency
SwiftFaceFormer-L3	36.9	27.1	28.0	2,015.6	**95.6**	2.6
SwiftFaceFormer-L1	18.0	55.3	11.8	804.6	95.0	5.3
SwiftFaceFormer-S	12.8	77.7	6.0	485.2	94.8	7.4
SwiftFaceFormer-XS	9.1	109.6	3.4	293.7	94.0	10.3
SwiftFaceFormer-XXS-KD	**4.6**	**215.5**	**1.5**	**64.1**	92.4	**20.1**

Table 5. Efficiency metrics in terms of latency, FPS throughput, number of parameters, and FLOPs tested on the Nvidia Jetson Nano platform. Our XXS-KD variant shows remarkable efficiency performance across all metrics.

5 Conclusion

This paper introduces SwiftFaceFormer, a novel family of hybrid models using Lightweight Face CNNs and Transformer architectures specifically tailored for face recognition tasks by adapting the SwiftFormer model and incorporating a Global Depth-Wise Convolution layer, followed by a 1×1 convolution layer and batch normalization to produce a compact 512-dimensional feature vector. Our most notable contribution is the design of our lightest version, SwiftFaceFormer-XXS, where we utilize grouped point-wise convolutions in specific sections of SwiftFormer's Convolution Encoders and progressively decrease the groups per stage for maximizing efficiency. Finally, we demonstrate that by using Knowledge Distillation for training our XXS variant, we are able to achieve remarkable balance between accuracy and efficiency for real-time resource-constrained environments.

Acknowledgment

This work was partially funded by the SOTERIA H2020 project. SOTERIA received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No101018342. This content reflects only the author's view. The European Agency is not responsible for any use that may be made of the information it contains.
The authors would like to thank the financial support from Tecnologico de Monterrey through the "Challenge-Based Research Funding Program 2022". Project ID # E120 - EIC-GI06 - B-T3 - D

References

1. Alansari, M., Hay, O.A., Javed, S., Shoufan, A., Zweiri, Y., Werghi, N.: Ghostfacenets: Lightweight face recognition model from cheap operations. IEEE Access (2023)
2. Boutros, F., Damer, N., Fang, M., Kirchbuchner, F., Kuijper, A.: Mixfacenets: Extremely efficient face recognition networks. In: 2021 IEEE International Joint Conference on Biometrics (IJCB). pp. 1–8. IEEE (2021)

3. Boutros, F., Siebke, P., Klemt, M., Damer, N., Kirchbuchner, F., Kuijper, A.: Pocketnet: Extreme lightweight face recognition network using neural architecture search and multistep knowledge distillation. IEEE Access **10**, 46823–46833 (2022)
4. Chen, S., Liu, Y., Gao, X., Han, Z.: Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In: Biometric Recognition: 13th Chinese Conference, CCBR 2018, Urumqi, China, August 11-12, 2018, Proceedings 13. pp. 428–438. Springer (2018)
5. Deng, J., Guo, J., Ververas, E., Kotsia, I., Zafeiriou, S.: Retinaface: Single-shot multi-level face localisation in the wild. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5203–5212 (2020)
6. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4690–4699 (2019)
7. Deng, J., Guo, J., Zhang, D., Deng, Y., Lu, X., Shi, S.: Lightweight face recognition challenge. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)
8. Elsken, T., Metzen, J.H., Hutter, F.: Neural architecture search: A survey. The Journal of Machine Learning Research **20**(1), 1997–2017 (2019)
9. George, A., Ecabert, C., Shahreza, H.O., Kotwal, K., Marcel, S.: Edgeface: Efficient face recognition model for edge devices. arXiv preprint arXiv:2307.01838 (2023)
10. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14. pp. 87–102. Springer (2016)
11. Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C.: Ghostnet: More features from cheap operations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1580–1589 (2020)
12. He, L., He, L., Peng, L.: Cformerfacenet: Efficient lightweight network merging a cnn and transformer for face recognition. Applied Sciences **13**(11), 6506 (2023)
13. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In: Workshop on faces in'Real-Life'Images: detection, alignment, and recognition (2008)
14. Kolf, J.N., Boutros, F., Elliesen, J., Theuerkauf, M., Damer, N., Alansari, M., Hay, O.A., Alansari, S., Javed, S., Werghi, N., et al.: Efar 2023: Efficient face recognition competition. arXiv preprint arXiv:2308.04168 (2023)
15. Li, J., Zhou, L., Chen, J.: Mobilefaceformer: a lightweight face recognition model against face variations. Multimedia Tools and Applications pp. 1–17 (2023)
16. Li, Y., Yuan, G., Wen, Y., Hu, J., Evangelidis, G., Tulyakov, S., Wang, Y., Ren, J.: Efficientformer: Vision transformers at mobilenet speed. Advances in Neural Information Processing Systems **35**, 12934–12949 (2022)
17. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV). pp. 116–131 (2018)
18. Maaz, M., Shaker, A., Cholakkal, H., Khan, S., Zamir, S.W., Anwer, R.M., Shahbaz Khan, F.: Edgenext: efficiently amalgamated cnn-transformer architecture for mobile vision applications. In: European Conference on Computer Vision. pp. 3–20. Springer (2022)
19. Martindez-Diaz, Y., Luevano, L.S., Mendez-Vazquez, H., Nicolas-Diaz, M., Chang, L., Gonzalez-Mendoza, M.: Shufflefacenet: A lightweight face architecture for efficient and highly-accurate face recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)

20. Martinez-Diaz, Y., Nicolas-Diaz, M., Mendez-Vazquez, H., Luevano, L.S., Chang, L., Gonzalez-Mendoza, M., Sucar, L.E.: Benchmarking lightweight face architectures on specific face recognition scenarios. Artificial Intelligence Review pp. 1–44 (2021)
21. Maze, B., Adams, J., Duncan, J.A., Kalka, N., Miller, T., Otto, C., Jain, A.K., Niggel, W.T., Anderson, J., Cheney, J., et al.: Iarpa janus benchmark-c: Face dataset and protocol. In: 2018 international conference on biometrics (ICB). pp. 158–165. IEEE (2018)
22. Moschoglou, S., Papaioannou, A., Sagonas, C., Deng, J., Kotsia, I., Zafeiriou, S.: Agedb: the first manually collected, in-the-wild age database. In: proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 51–59 (2017)
23. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4510–4520 (2018)
24. Sengupta, S., Chen, J.C., Castillo, C., Patel, V.M., Chellappa, R., Jacobs, D.W.: Frontal to profile face verification in the wild. In: 2016 IEEE winter conference on applications of computer vision (WACV). pp. 1–9. IEEE (2016)
25. Shaker, A., Maaz, M., Rasheed, H., Khan, S., Yang, M.H., Khan, F.S.: Swiftformer: Efficient additive attention for transformer-based real-time mobile vision applications. arXiv preprint arXiv:2303.15446 (2023)
26. Su, W., Wang, Y., Li, K., Gao, P., Qiao, Y.: Hybrid token transformer for deep face recognition. Pattern Recognition **139**, 109443 (2023)
27. Tan, M., Le, Q.V.: Mixconv: Mixed depthwise convolutional kernels. arXiv preprint arXiv:1907.09595 (2019)
28. Tang, Y., Han, K., Guo, J., Xu, C., Xu, C., Wang, Y.: Ghostnetv2: enhance cheap operation with long-range attention. Advances in Neural Information Processing Systems **35**, 9969–9982 (2022)
29. Wang, M., Deng, W.: Deep face recognition: A survey. Neurocomputing **429**, 215–244 (2021)
30. Whitelam, C., Taborsky, E., Blanton, A., Maze, B., Adams, J., Miller, T., Kalka, N., Jain, A.K., Duncan, J.A., Allen, K., et al.: Iarpa janus benchmark-b face dataset. In: proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 90–98 (2017)
31. Yan, M., Zhao, M., Xu, Z., Zhang, Q., Wang, G., Su, Z.: Vargfacenet: An efficient variable group convolutional neural network for lightweight face recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)
32. Zhang, Q., Li, J., Yao, M., Song, L., Zhou, H., Li, Z., Meng, W., Zhang, X., Wang, G.: Vargnet: Variable group convolutional neural network for efficient embedded computing. arXiv preprint arXiv:1907.05653 (2019)
33. Zheng, T., Deng, W.: Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. Beijing University of Posts and Telecommunications, Tech. Rep **5**(7) (2018)
34. Zheng, T., Deng, W., Hu, J.: Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. arXiv preprint arXiv:1708.08197 (2017)
35. Zhong, Y., Deng, W.: Face transformer for recognition. arXiv preprint arXiv:2103.14803 (2021)