IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 22 March 2023, accepted 5 April 2023, date of publication 10 April 2023, date of current version 13 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3266068

RESEARCH ARTICLE

GhostFaceNets: Lightweight Face Recognition Model From Cheap Operations

MOHAMAD ALANSARI[1], (Member, IEEE), OUSSAMA ABDUL HAY[2,3], SAJID JAVED[1,4], ABDULHADI SHOUFAN[1,4,5], YAHYA ZWEIRI[2,3], (Member, IEEE), AND NAOUFEL WERGHI[1,4,5], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, Khalifa University, Abu Dhabi, United Arab Emirates
[2]Department of Aerospace Engineering, Khalifa University, Abu Dhabi, United Arab Emirates
[3]Advanced Research and Innovation Center (ARIC), Khalifa University, Abu Dhabi, United Arab Emirates
[4]Center for Autonomous Robotic Systems, Khalifa University, Abu Dhabi, United Arab Emirates
[5]Center for Cyber-Physical Systems (C2PS), Khalifa University, Abu Dhabi, United Arab Emirates

Corresponding author: Mohamad Alansari (100061914@ku.ac.ae)

This work was supported by the Khalifa University of Science and Technology under Award RC1-2018-KUCARS.

ABSTRACT The development of deep learning-based biometric models that can be deployed on devices with constrained memory and computational resources has proven to be a significant challenge. Previous approaches to this problem have not prioritized the reduction of feature map redundancy, but the introduction of Ghost modules represents a major innovation in this area. Ghost modules use a series of inexpensive linear transformations to extract additional feature maps from a set of intrinsic features, allowing for a more comprehensive representation of the underlying information. GhostNetV1 and GhostNetV2, both of which are based on Ghost modules, serve as the foundation for a group of lightweight face recognition models called GhostFaceNets. GhostNetV2 expands upon the original GhostNetV1 by adding an attention mechanism to capture long-range dependencies. Evaluation of GhostFaceNets using various benchmarks reveals that these models offer superior performance while requiring a computational complexity of approximately 60-275 MFLOPs. This is significantly lower than that of State-Of-The-Art (SOTA) big convolutional neural network (CNN) models, which can require hundreds of millions of FLOPs. GhostFaceNets trained with the ArcFace loss on the refined MS-Celeb-1M dataset demonstrate SOTA performance on all benchmarks. In comparison to previous SOTA mobile CNNs, GhostFaceNets greatly improve efficiency for face verification tasks. The GhostFaceNets code is available at: https://github.com/HamadYA/GhostFaceNets.

INDEX TERMS ArcFace, attention mechanism, cheap operations, face recognition, GhostNet, lightweight.

I. INTRODUCTION

Over the past few years, accessing information through smartphones and tablets has become commonplace in both professional and private settings. Mobile devices have become indispensable tools in our daily lives as the use of these devices for services like social networks, email, electronic commerce, and banking has surpassed that of traditional computers. Users and corporations may be subject to security concerns and threats without the proper security options [1]. Vision-based tasks such as Face Detection (FD), Face Recognition (FR), and Face Verification (FV) are commonly used as an authentication option for protection purposes in the smartphone [2]. Deep learning-based approaches have been observed to deliver more satisfactory results and improve State-of-The-Art (SOTA) compared to traditional 'shallow' schemes in most vision-based tasks [3], [4], specifically in FR and FV tasks [5]. However, deploying FR deep learning-based models on embedded domains such as mobile devices is constrained by the computational resources and the high throughput requirements [6], [7], [8], since FR deep learning-based models rely on a huge number of parameters [9], [10].

Recent developments in FR showed great progress in overcoming these limitations. Some approaches utilized pre-trained SOTA FR to transfer knowledge from big to small

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

model by using Knowledge Distillation (KD) [11]. Others used the quantization techniques to quantize their model, hence reducing its size [12], and low-rank approximation which effectively reduces computations [13]. Creating lightweight deep neural networks has emerged as one of the most promising ways to improve speed-accuracy trade-offs in recent years [14], [15], [16], [17], [18]. We use the lightweight attribute to describe models that show computational complexity in the range of 1G floating point operations (FLOPs). SqueezeNet [14], MobileNets [15], ShuffleNets [16], [17], VarGNet [18], and MixNets [19] are examples of such networks that provide promising results in image classification. Few works have proposed utilizing such lightweight deep learning architectures as a backbone for FR models. For example, MobileFaceNet [20], ShuffleFaceNet [21], VarGFaceNet [22], and MixFaceNets [23] employed MobileNetV2 [15], ShuffleNetV2 [17], VarGNet [18], and MixNets [19] as FR model's backbone respectively. Very recently, extremely lightweight backbones namely GhostNetV1 and GhostNetV2 were proposed for image classification tasks [24], [25]. GhostNetV1 proposed a novel Ghost module that generates more features by using fewer parameters, which were then extended in GhostNetV2 to incorporate long-range dependencies too. Experiments showed that GhostNetV1 and GhostNetV2 frequently beat competitors at various degrees of computational complexity [24]. GhostNetV1 specifically outperformed MobileNets [15], ShuffleNets [16], [17], and many other models on image classification tasks while GhostNetV2 outperformed the first version.

In this work, we propose a new family of lightweight architectures named GhostFaceNets that adopts GhostNetV1 and GhostNetV2 (referred to as GhostNets in this paper) [24] as backbones in the FR field. We carefully designed the output layer's head termed modified Global Depthwise Convolution (GDC) to be suitable for FR and FV tasks. Firstly, we adjusted the Squeeze and Excitation (SE) module [14] aiming at improving the discriminative power of GhostFaceNets. Secondly, we replaced Rectified Linear Unit (ReLU) in the GhostNets architectures with Parametric Rectified Linear Unit (PReLU) as a nonlinear activation function since the latter provides higher accuracy compared to the former [26], [27]. Finally, we adopted the ArcFace loss function for the feedback signal thanks to its effective enhancement of intra-class compactness and inter-class discrepancy [9]. As a result, our GhostFaceNets achieved SOTA on the most commonly used validation benchmarks, with a computation complexity of approximately between 60 MFLOPs and 275 MFLOPs based on the width and strides (discussed in Section IV). Also, our GhostFaceNets achieved comparable results to the SOTA compact FR/FV models that have an extremely higher computation complexity.

The paper is organized as follows. Section II provides an overview of the existing lightweight models for FR/FV. Section III briefly explains the SOTA GhostNetV1 and GhostNetV2 architectures. Section IV illustrates in detail the GhostFaceNets architecture proposed for FR. Section V explains the experimental setup. Section VI shows the results, and Section VII concludes our work.

II. RELATED WORK

Deep learning methods such as Convolutional Neural Networks (CNNs) have revolutionized and reshaped the FR research landscape in all aspects, achieving a huge increase in accuracy compared to "shallow" methods [5]. However, these networks offer a poor trade-off between performance and model complexity [9], [10]. A good trade-off between performance and model complexity is a challenge for the FR community and a requirement in real-world applications and embedded devices [6], [7], [8]. In this section, we give a brief overview of the most recent advances in developing lightweight deep learning models for FR. Furthermore, we summarize (as long as available) the computational complexity and FR accuracy on the Labeled Faces in the Wild (LFW) dataset [28]. LFW is the most used dataset for reporting the accuracy of new FR models [29].

A CNN model was proposed in [30] as an attempt to develop an efficient lightweight FR model. The authors proposed an architecture to learn a compact embedding on the massively noisy labels in large-scale face data. They designed three lightweight networks (Light CNN-4, Light CNN-9, and Light CNN-29) with the aim of reducing the number of parameters and computational complexity. The authors used Max-Feature-Map activation after each CNN convolutional layer and proposed using a bootstrapping technique to overcome the noisy labels issue. The best-reported accuracy on the LFW dataset of the three networks was 99.33% with 12.637M parameters and around 3.9 GFLOPs [30]. Thus, these models are considered computationally expensive and not suitable for embedded devices.

In [31], the authors introduced a parameter- and FLOPs-free "Shift" operation as a replacement for spatial convolutions. The authors used the FaceNet [32] architecture that is based on Inception-Resent V1 [33], [34] which contains around 28.5M parameters and 1.6 GFLOPs. The proposed ShiftFaceNet reduced the number of parameters by approximately 36.54 times. However, this architecture caused a drop in accuracy of around 2 degrees compared to the original FaceNet [31], [32].

The models that rely on image classification backbones were then introduced to offer a better trade-off between performance and model complexity [20], [21], [22], [23]. MobileFaceNets [20] employed MobileNetV2 [15] for high accuracy and real-time FR and FV on mobile and embedded devices. The authors finetuned the MobileNetV2 architecture by using PReLU [27] as the nonlinear activation function and replacing the Global Average Pooling (GAP) layer with a GDC layer which demonstrated to provide more discriminative face representation [20], [21]. The model outperformed many FR/FV SOTA models on the LFW dataset achieving a best-reported accuracy of 99.55% with only around 1M

parameters, 439.7 MFLOPs, and 4.0 MB model size, making this model suitable for real-time embedded systems [20].

ShuffleFaceNet [21], proposed to improve the accuracy of previous lightweight models, followed the same procedure as in [20] by utilizing ShuffleNetV2 [17] as a backbone and finetuning this by using GDC instead of the GAP layer and PReLU as a nonlinear activation function [27]. The authors designed four variants of the ShuffleFaceNet models with different complexity levels. The best-reported accuracy on the LFW dataset was 99.67% with 2.6M parameters, 557.5 MFLOPs, and a model size of 10.5 MB using the variant ShuffleFaceNet 1.5× utilizing ArcFace loss function [9], [21]. ShuffleFaceNet is slightly larger than MobileFaceNet in terms of complexity, parameters, and model size; however, it offers better accuracy [20], [21].

Following a similar pattern as in [20] and [21], VarGFaceNet [22] employed VarGNet [18] as its backbone. In contrast to [20] and [21], the authors proposed a new embedding block by first adding a SE module [14] and the PReLU nonlinear activation function [27] to improve discriminative ability. Secondly, the downsampling in VarGNet [18] is removed to preserve face information. Third, variable group convolution is used before the Fully Connected (FC) layer to decrease the parameters. Finally, a recursive KD was proposed to improve the generalization gap with SOTA FR models. KD is the procedure of transferring knowledge from a teacher (a deep neural network with high performance and complexity) to a student (a small model with low complexity) aiming to transfer the large knowledge capacity of the teacher to the student to improve its performance [35]. The model with recursive KD achieved 99.85% accuracy on the LFW dataset with around 5M parameters and 1.022 GFLOPs [22]. Although the model outperformed all SOTA FR models; however, its computational cost remained higher than MobileFaceNet, ShuffleFaceNet [20], [21], [22].

Most recently, a new family of lightweight efficient FR models was proposed named MixFaceNets [23]. Almost similar to [20] and [21], MixFaceNets employed MixNets [19] as a backbone. To improve the discriminative ability of MixNets, the authors modified the MixConv block [19] with a channel shuffle operation [17]. The best-reported accuracy on the LFW dataset was 99.68% using 3.95M parameters and 626.1 MFLOPs, which makes it highly efficient for real-time embedded systems [23].

Due to the improved trade-off between performance and model complexity, the MobileFaceNet architecture [20] was adopted in a new FR model called AirFace [36]. The authors modified the architecture by adding the convolutional block attention module [36] to every bottleneck in the architecture. In addition, the authors modified the ArcFace loss function [9] by replacing the cosine function with a linear function and introduced a new loss function named Li-ArcFace [36]. The proposed Li-ArcFace loss function showed a higher convergence when training the model with a small embedding size as compared with ArcFace [9], [36]. The best-reported accuracy on the LFW dataset was 99.27% with 1 GFLOPs, which makes the model expensive in terms of computational complexity [36].

The model size, measured in MB, is a concern for real-time embedded systems compatibility [6], [7], [8]. Quantization methods have been shown to reduce the size of the model as in [12]. QuantFace is probably the first model to use quantization in the FR domain [12]. To avoid reducing the accuracy, the authors also proposed using KD which modifies the quantized model and its parameters using synthetically created face data by Generative Adversarial Networks (GANs) [37]. Considering quantization applied on MobileFaceNet model [20], the best-reported accuracy on the LFW dataset was 99.43% with 1.1M parameters and model size of 1.1 MB [12].

A set of lightweight FR models, dubbed PocketNets, was proposed in [11]. PocketNets utilized Neural Architecture Search (NAS) [38], [39] to automatically create efficient artificial neural networks. NAS automates the process of a human designing a neural network and learning what works effectively [38]. In addition, the authors proposed a novel KD paradigm aiming to ease the challenges caused by the significant gap between the teacher and student models [11]. The authors successfully achieved an improved trade-off between model performance and compactness, achieving 99.58% accuracy on the LFW dataset with only 0.925M parameters and 587.11 MFLOPs [11].

In this work, we propose a new set of lightweight architectures named GhostFaceNets, that extends two efficient neural architectures, named GhostNetV1 and GhostNetV2 (we refer to them as GhostNets in this paper) [24] to the field of FR and FV. First, we removed the GAP layer, the pointwise convolution layer (1×1 convolution layer), and the FC layer and replaced them with our proposed modified GDC recognition head. Second, we replaced ReLU, which is used in GhostNets, with PReLU as a nonlinear activation function because PReLU eliminates the problem of the vanishing gradient and its performance improvement over ReLU [26], [27]. Third, the conventional FC layers in the SE modules were replaced by convolution layers to improve the discriminative power of GhostFaceNets [14]. Finally, we employed the ArcFace loss function as a feedback signal used for training [9]. We choose the ArcFace loss function because it achieved a superior accuracy boost when used with FR / FV models, since it enforces intra-class compactness, inter-class discrepancy, classification margin, and enhances the discriminative power of learned features [9]. Moreover, we experimented with the performance of the proposed GhostFaceNets under different hyperparameters settings shown in Section V-C. As a result, we designed a set of GhostFaceNets models by changing the training dataset, the width of GhostNets architectures, and the stride of the first convolution layer (referred to as the stem of the model). The results show that GhostFaceNets outperforms most lightweight SOTA models on all validation/testing benchmarks as discussed in

Section VI. The major contributions of this work are summarized as follows:

- With two different levels of complexity, we create lightweight FR architectures that are accurate and efficient. The resulting GhostFaceNets models are suited for deployment on real-time applications as well as mobile and embedded devices. We show that the models have an actual model size of less than 13.8 MB and an actual inference CPU time of roughly 50 ms using the TFLite tool.
- We designed a modified GDC layer to generate a discriminative feature vector and PReLU as a nonlinear activation function to assure not only speed and little storage space but also notable gains on FR/FV accuracy.
- We adjusted the SE module by replacing the conventional FC layers in the SE modules with convolution layers to improve GhostFaceNets' discriminative power.
- We show that our GhostFaceNets outperform SOTA CNNs on widely used FR/FV benchmarks.
- We determined the best model hyperparameters that offer the best accuracy to computational complexity trade-off by undergoing an extensive ablation study in Section V-C.
- We determined the best loss function among three well-known loss functions in FR/FV field in Section V-C. We believe that this will ease the choice of the loss function for researchers when developing a new FR model.

III. PRELIMINARIES

In this section, we briefly explain the SOTA GhostNetV1 and GhostNetV2 that inspired our work for a better understanding of GhostFaceNets.

A. GHOST MODULES – FEATURE MAP PATTERN REDUNDANCY

In GhostNetV1, Ghost modules are employed to generate a certain percentage, denoted as $x\%$, of the feature maps, while the remaining feature maps are generated using a low-cost linear operation known as depthwise convolution (DWConv). The resulting tensor of feature maps has C' channels. This approach differs from a traditional convolutional layer, which generates a tensor of feature maps directly from an input tensor of C channels, with C' channels in the resulting tensor. In particular, a 2D filter, i.e., kernel, is applied to a 2D channel of the input tensor to generate a 2D channel of the output tensor. This drastically reduces the number of parameters and FLOPs without considerable impact on the performance. The linear operation mimics intrinsic convolution in terms of features. So, it can be learned from the input using back-propagation in the backward pass. Note that the number of channels in the input tensor and the resulting output tensor in the specific layer must match for depthwise convolutions to improve speed and decrease complexity.

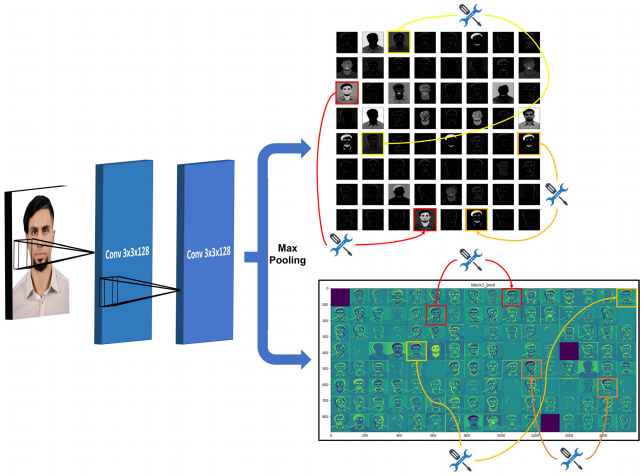

FIGURE 1. Visualizing feature maps generated by the first block of VGG16 using two visualization settings. Very similar feature maps (annotated with boxes of the same color) are generated and visualized using gray visualization and Viridis visualization, which support the idea of generating feature maps of one feature map using linear transformations (cheap operations denoted by spanners and screwdrivers).

Many works designed a deep CNN [40], [41], [42] to effectively address the image classification task. However, these attempts have a poor trade-off between model performance and compactness. Attempts were made to improve the trade-off [15], [16], [17], [18], [19] by introducing operations such as shuffle operation and depthwise convolutions, which have significantly improved the trade-off. However, the use of 1×1 convolutions adds additional computational complexity. Previous works [40], [41], [42] relied on the fact that the deeper CNN is, the more features it will generate, and thus the better the performance will be.

Ghost modules exploit the observation that multiple identical copies of unique intrinsic feature maps, which would otherwise require computationally expensive convolutional operations, can be identified within the set of feature maps generated by the convolutional layer [24]. Taking the output of the first block of VGG16 [40] as an example in Figure 1. We used two visualization settings, namely the gray visualization at the top displaying 8×8 feature maps, and the Viridis visualization at the bottom displaying 8×16 feature maps. As shown below, there are clearly similar and redundant feature map pairs (we can keep only one pair) detonated as *ghosts* which can be generated using linear operations. Furthermore, it can be observed that there is sparsity in the output obtained from the Viridis visualization settings, as shown in Figure 1. This observation indicates that certain neurons are inactivated and consequently, not useful. The results in Figure 1 stimulate the idea of generating other feature maps from a single feature map using cheap operations which reduces the computational complexity of the network. We believe that these similar and redundant features denoted in colored boxes in Figure 1 are crucial for a high-performing CNN. Hence, these similar features are generated using cheap operations rather than discarding them.

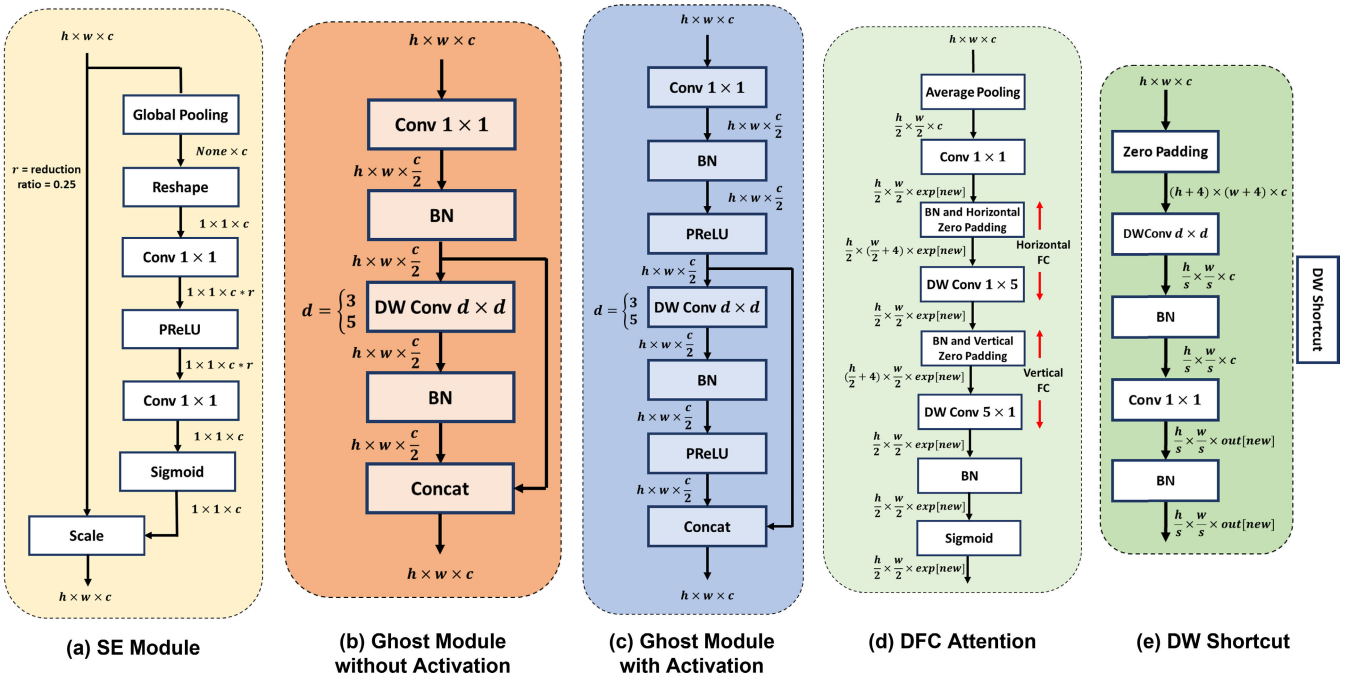

FIGURE 2. (a) SE Module. (b) Ghost module structure without PReLU activation function. (c) Ghost module structure with PReLU activation function. (d) DFC attention module. (e) Depthwise shortcut.

Based on the above, GhostNetV1 authors proposed to replace the convolution layers in deep neural network designs with an essentially freestanding replacement layer named Ghost module. In Ghost modules, the output tensor for every convolutional layer is formed by serialization of two processes. 1) Create the first $x\%$ of the total channels of the output tensor using a sequential stack of three layers that includes standard convolution, batch normalization [43], and a nonlinear activation function, which is by default specified as a ReLU [26]. 2) The output from this is then sent to the second block, which is once more a sequential stack of three layers consisting of depthwise convolution, batch normalization [43], and ReLU [26]. The output tensor is completed by stacking the tensor from the first sequential block with the output from the secondary sequential block.

Using the above fact, the Ghost module can effectively generate the same number of feature maps as the ordinary convolution layer. Therefore, it can be easily integrated (added) into any existing neural networks such as [15], [16], [17], [18], [19], [40], [41], and [42] to reduce computational complexity. The structure of the Ghost module with and without the activation function is depicted in Figure 2 (b) and (c).

B. GhostNetV1

Using Ghost modules, a novel backbone architecture called GhostNetV1 was proposed in [24] which is effectively a modified version of MobileNetV3 [44] with a Ghost bottleneckV1 in place of the former bottleneck. These Ghost bottlenecksV1 are essentially made up of Ghost modules which have the same architectural design as a typical MobileNetV3 bottleneck [24], [44] as shown in Figure 3 (a), (b), (c), and (d).

After the input layer, which is a typical convolutional layer (denoted as the stem of the model), GhostNetV1 is constructed by stacking Ghost bottlenecksV1 with increasing channels in the tensor in succession. Based on the dimensionality of the input feature map, a staged grouping of the Ghost bottlenecksV1 is created. Except for the last bottleneckV1, where the stride 2 design was employed, all the Ghost bottlenecksV1 were applied with a stride of 1. The SE modules [14] were also utilized in [24] to offer channel attention for a few remaining connections in the Ghost bottlenecksV1, increasing accuracy with a minimal computational cost. SE modules are applied to the residual layer in some Ghost bottlenecksV1 with a SE ratio $r = 0.25$ [14], [24].

GhostNetV1 latency speed, model size, computational cost, and accuracy are controlled by the width multiplier factor denoted as α. A width multiplier can roughly regulate the model size and computational cost by a factor of α^2. Smaller α leads to low performance and computational cost and vice versa. A GhostNetV1 with a width multiplier is denoted as GhostNetV1-α.

The hyperparameters of the GhostNetV1 architecture are described in detail in Appendix Section-A. These hyperparameters are important to understand the proposed GhostFaceNets.

C. GhostNetV2

Drawing inspiration from attention-based models [45], [46], [47], [48], the authors of GhostNetV2 proposed to

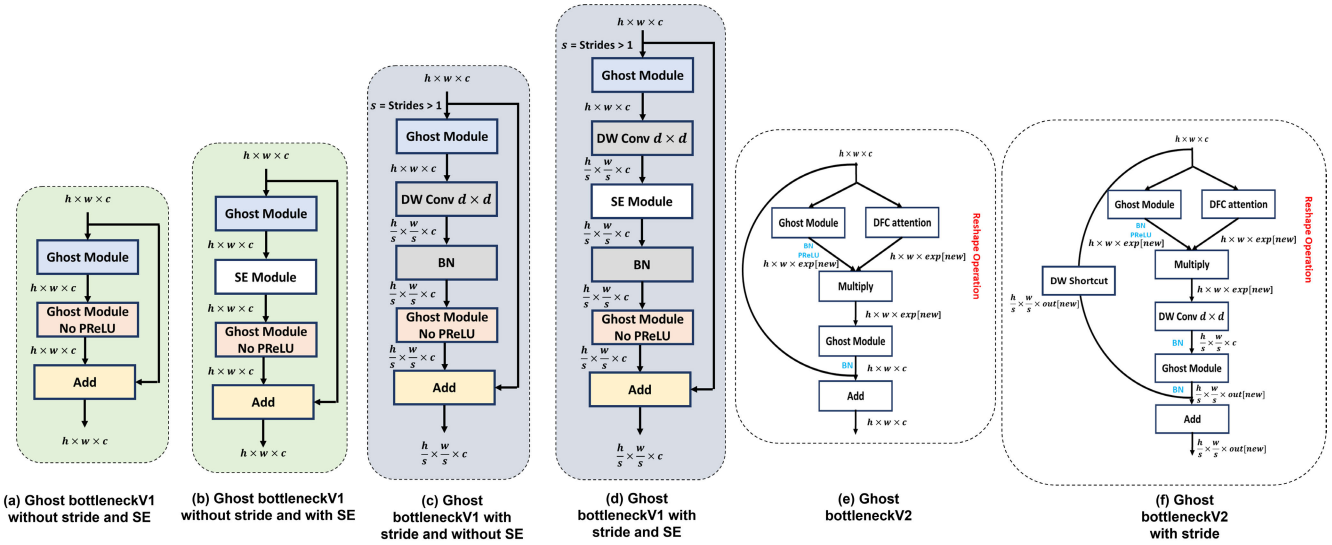

FIGURE 3. (a) Ghost bottleneckV1 structure without stride and SE. (b) Ghost bottleneckV1 structure with SE but without stride. (c) Ghost bottleneckV1 structure with stride and SE. (d) Ghost bottleneckV1 structure with stride but without SE. (e) Ghost bottleneckV2 structure with stride.

enhance the Ghost module of GhostNetV1 by incorporating long-range dependencies [24], [25]. To achieve this, they introduced a novel attention-based layer named the DFC attention branch, shown in Figure 2 (d), which utilizes convolutions to generate attention maps with global receptive fields [25]. Unlike the self-attention layers employed in MobileViT [49], [50], the DFC attention branch is highly efficient and capable of capturing long-range dependencies between pixels located in different spatial locations. It is noteworthy that many prior attention modules are hardware-unfriendly due to the extensive use of tensor reshaping and transposing operations to implement feature splitting and attention calculation [25]. To ensure hardware compatibility and avoid slowing the inference time, the authors aimed to develop an attention-based module that is computationally efficient and requires minimal tensor operations.

Based upon the DFC attention branch, a new bottleneck is created shown in Figure 3 (e) and (f). The Ghost module and the DFC attention are two parallel branches, taking the same input and collecting information from various viewpoints, causing an information aggregation procedure. Their element-wise product, which incorporates data from both the Ghost module's features and the DFC attention module's attentions, is the output. Each attention value is calculated using patches over a wide range so that the output feature can incorporate data from these patches [24], [25].

The DFC attention branch, shown in Figure 2 (d), consists of five operations: 1) downsample, 2) Convolution, 3) Horizontal FC, 4) Vertical FC, and 5) Sigmoid. To reduce the extra computational cost of directly paralleling the DFC attention module with the Ghost module, a native average pooling and bilinear interpolation for downsampling and upsampling are used, respectively. The horizontal FC consists of a batch normalization layer and zero padding in the horizontal direction (add 4 pixels to the width), followed by depthwise convolution in the horizontal direction to remove the newly added 4 pixels. And the same structure for the vertical FC but in the vertical direction (height). Decomposing the FC layer into horizontal FC and vertical FC has been shown to reduce computational complexity compared to conventional FC layers and, at the same time, capture long-range dependencies along the two directions [25].

The hyperparameters of the new bottleneck architecture are discussed in Appendix Section-A.

IV. GhostFaceNets

This section provides an explanation of the proposed lightweight FR/FV models called GhostFaceNets. These models draw inspiration from the SOTA GhostNets [24], [25], see Figure 4. Three main modifications are proposed:

- Applying different output head settings (named modified GDC) shown in Figure 5.
- Replacing ReLU by PReLU as our networks' activation function.
- Adjusting the SE modules to improve the discriminative power of GhostFaceNets, see Figure 2 (a).
- Employing the ArcFace loss function which is chosen based on an extensive ablation study in Section V-C.

Most deep networks designed for image classification, including GhostNets [24], [25], use the output of the GAP layer as a feature vector in the embedding process. However, this method has proven to be less successful when used for FR/FV [9], [20], [30]. This is because the GAP layer treats each unit of the output feature map equally, which conflicts with the assumption that different types of units bring different amounts of discriminative information to the theory when it comes to extracting a face feature vector. Instead, we can learn various weights for these units using an FC

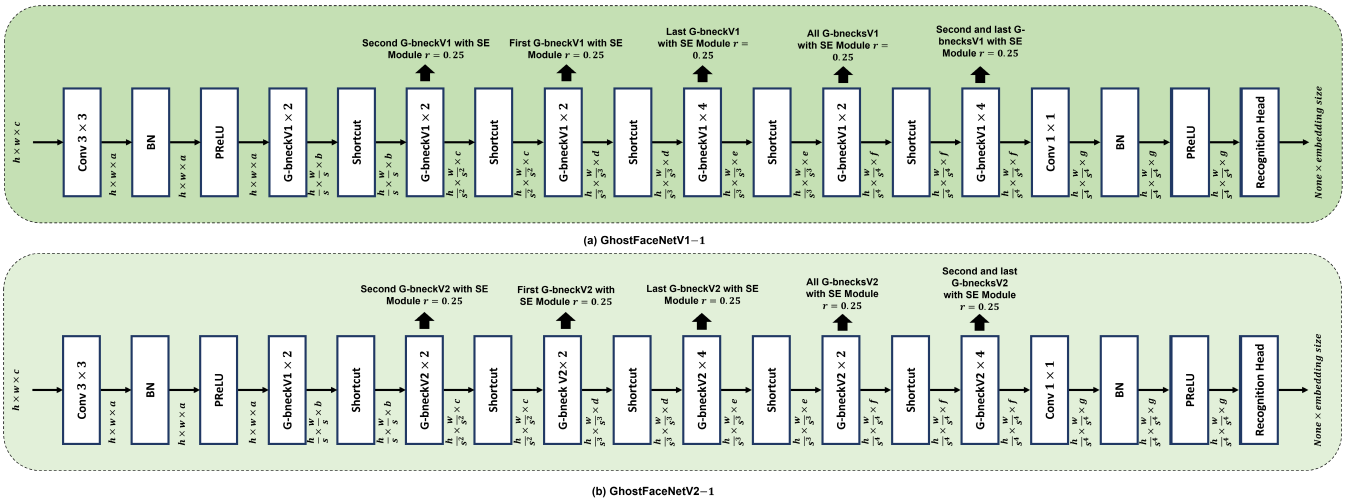

FIGURE 4. (a) Proposed GhostFaceNetV1-1 architecture. (b) Proposed GhostFaceNetV2-1 architecture.

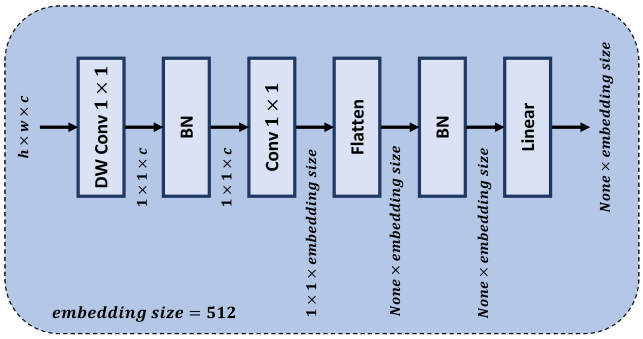

FIGURE 5. Modified GDC, Recognition Head.

layer, then project the knowledge into a small facial feature vector. Nevertheless, the FC layer ended up having a lot of weights, which not only makes the model bigger but also calls for additional processing power. A GDC layer was recently employed in [20] to treat various output feature map units with varying degrees of relevance, demonstrating that it is an effective structure for FR.

In this work, we modified the GDC layer and replaced the GhostNets GAP layer with the modified GDC layer. The modified GDC shown in Figure 5 consists of a GDC layer of 7 × 7 kernel size followed by batch normalization layer [43]. A convolution layer of size $1 \times 1 \times embeddingsize$ is then applied to create the desired embedding vector (size). A flattening layer, batch normalization layer [43], and a linear activation function are then added to the top to produce a compact 512-dimensional embedding vector (size).

GhostNets adopt the ReLU activation function, which only permits non-negative activations, as their nonlinearity activation function [24], [25], [26]. However, various activation functions have been proposed to address this limitation [27], [30], [36], [51], [52]. In this study, we opted to use the PReLU activation function [27] over ReLU [26] inspired

Algorithm 1 |_MAKE_DIVISIBLE(V, D, MIN_VALUE)

Input: *V*: Input 1, *D*: divisor, *min_value*
Output: *new_v*

```
if min_value is None:
    | min_value = divisor
new_v = max ( min_value, int (v + divisor / 2) // divisor * divisor)
if new_v < 0.9 * v:
    | new_v += divisor
return new_v
```

by related work and confirmed its empirically superior performance in FR tasks [20], [21], [22], [23]. PReLU enables negative activations, enhancing the network's ability to learn complex nonlinear functions, ultimately improving network performance.

Additionally, conventional FC layers in SE modules [14] are replaced by convolutions with the setup shown in Figure 2 (a). We use GAP followed by a reshaping operation to convert the output shape to the format $1 \times 1 \times channel\_axis$ so that convolution can be applied to it. Then a convolution is applied to the reshaped output with 1 × 1 kernel size and reduction factor filters. The reduction factor is formulated as,

$$\_make\_divisible(channel\_axis \times r) \approx channel\_axis \times r, \tag{1}$$

where r is the SE ratio, and *_make_divisible* is defined in Algorithm 1 as pseudo code. The convolution is followed by the PReLU activation function, which ends the squeeze operation. A second convolution of kernel size 1×1 and *channel_axis* filter followed by the sigmoid activation function is then applied for the excitation operation. The excitation operation basically multiplies the *channel_axis* by $1/r$ reshaping it back to $1 \times 1 \times channel\_axis$. The output of the sigmoid activation function is then rescaled by multiplying it by the input of the SE module. We believe that using channel-wise

TABLE 1. The performance of the proposed GhostfacenetV1-2 with different α. The best performance in each benchmark is in bold.

Model	α	Com-plexity FLOPs	# Params.	LFW (%)	AgeDB_30 (%)	CFP_FP (%)
GhostFaceNet V1-2	0.5	10,785,302	718,484	98.783	91.13	83.4
GhostFaceNet V1-2	1	36,837,162	2,488,580	99.383	94.85	88.81
GhostFaceNet V1-2	1.2	51,832,732	3,508,496	99.53	95.62	90.13
GhostFaceNet V1-2	1.3	59,493,840	4,056,312	**99.65**	**96.18**	**91.51**
GhostFaceNet V1-2	1.4	68,911,476	4,662,132	99.6167	95.78	90.8857

TABLE 2. The perfromance of the proposed GhostFaceNetV1-2 with different loss functions. The best performance in each benchmark is in bold.

Model	α	Loss Function	LFW (%)	AgeDB_30 (%)	CFP_FP (%)
GhostFaceNet V1-2	1.3	ArcFace	**99.65**	**98.783**	**91.13**
GhostFaceNet V1-2	1.3	CosFace	99.5167	95.23	90.37
GhostFaceNet V1-2	1.3	Sub-center ArcFace	99.633	95.6	90.73
GhostFaceNet V1-1	1.3	ArcFace	**99.7667**	**97.58**	**95.64**
GhostFaceNet V1-1	1.3	CosFace	99.7167	97.1833	94.7429
GhostFaceNet V1-1	1.3	Sub-center ArcFace	99.667	95.85	91.46

attention mechanism configuration will further improve the channel interdependencies at almost no computational cost. The network includes a parameter that adjusts each channel's weight in such a way that it is more responsive to important features and less sensitive to unimportant features.

Lastly, we set a stride (*st*) hyperparameter that controls the stride hyperparameter of the model's stem (first convolution in the network) that gives the option of applying a fast downsampling strategy at the beginning of the network or not. If $st = 2$, then the fast downsampling is activated with a stride of 2.

For generalization, we designed GhostFaceNets variants by changing: 1) the backbone, 2) the width multiplier α, 3) and the stem of the model stride (*st*) hyperparameter. We then extensively studied the effect of different hyperparameters and training datasets; particularly, we used two training datasets (MS1MV2 and MS1MV3) in an ablation study in Section V-C. We denote GhostFaceNetVi $i \in \{1, 2\}$ of stride *st* and training dataset data as GhostFaceNeti-st (MS1MV*k*) $k \in \{2, 3\}$. We used strides 1 and 2 for the stride (*st*) hyperparameter creating GhostFaceNet*i*-1 (MS1MV*k*) presented in Figure 4 and detailed in Appendix Section-B and GhostFaceNet*i*-2 (MS1MV*k*). A GhostNets of width multiplier $\alpha = 1.3$ (determined by ablation study in Section V-C) is kept the same in the architectures which are shown to provide the best trade-off between performance and model complexity.

TABLE 3. The performance of the proposed GhostfacenetV2-2 with different α. The best performance in each benchmark is in bold.

Model	α	Training Dataset	Com-plexity FLOPs	# Params.	LFW (%)	AgeDB_30 (%)	CFP_FP (%)
GhostFace NetV2-2	1	MS1MV2	47,371,312	4,229,744	95.65	96.212	92.9
GhostFace NetV2-2	1.1	MS1MV2	56,262,058	5,033,078	99.6	95.683	91.814
GhostFace NetV2-2	1.2	MS1MV2	66,543,012	5,917,412	99.633	95.9	92.686
GhostFace NetV2-2	1.3	MS1MV2	76,513,640	6,839,104	**99.717**	**96.55**	93.071
GhostFace NetV2-2	1.4	MS1MV2	88,378,922	7,847,302	99.683	95.983	91.986
GhostFace NetV2-2	1.5	MS1MV2	100,386,924	8,927,420	99.7	96.483	**93.9**
GhostFace NetV2-2	1	MS1MV3	47,371,312	4,229,744	99.667	96.417	**94.857**
GhostFace NetV2-2	1.1	MS1MV3	56,262,058	5,033,078	99.667	96.167	92.471
GhostFace NetV2-2	1.2	MS1MV3	66,543,012	5,917,412	99.683	96.383	94.343
GhostFace NetV2-2	1.3	MS1MV3	76,513,640	6,839,104	**99.683**	**96.833**	94.286
GhostFace NetV2-2	1.4	MS1MV3	88,378,922	7,847,302	99.683	96.417	94.629
GhostFace NetV2-2	1.5	MS1MV3	100,386,924	8,927,420	99.667	96.417	**94.857**

TABLE 4. The performance of the proposed GhostFaceNetV2-2 with different loss functions. The best performance in each benchmark is in bold.

Model	α	Loss Function	LFW (%)	AgeDB_30 (%)	CFP_FP (%)
GhostFaceNet V2-2	1.3	ArcFace	**99.717**	**96.55**	**93.071**
GhostFaceNet V2-2	1.3	CosFace	99.5846667	95.6	91.931
GhostFaceNet V2-2	1.3	Sub-center ArcFace	99.73	95.97	92.291

TABLE 5. The performance of the proposed GhostFaceNets with different recognition heads. The best performance in each benchmark is in bold.

Model	α	Recognition Head	LFW (%)	AgeDB_30 (%)	CFP_FP (%)
GhostFaceNet V1-2	1.3	Modified GDC	**99.65**	**96.18**	**91.51**
GhostFaceNet V1-2	1.3	GDC	99.5867	95.65	91.26
GhostFaceNet V2-2	1.3	Modified GDC	**99.717**	**96.55**	**93.071**
GhostFaceNet V2-2	1.3	GDC	99.6167	95.94	91.21

The ArcFace loss function [9] is chosen as our training loss function since it minimizes intra-class gap and exhibits clear inter-class differentiation, outperforming other

TABLE 6. The performance of the proposed GhostFaceNets with different training datasets. The best performance in each benchmark is in bold.

Model	α	Training Dataset	LFW (%)	CA-LFW (%)	CP-LFW (%)	CFP-FP (%)	CFP-FF (%)	AgeDB _30 (%)	IJB-B (%)	IJB-C (%)	VGG2 -FP (%)
GhostFaceNetV1-1	1.3	MS1MV2	**99.7 ± 0.201**	95.84 ± 0.12	**91.9 ± 0.1**	95.9 ± 0.12	99.77 ± 0.1	97.55 ± 0.09	92.15 ± 0.123	94.03 ± 0.084	93.61 ± 0.09
GhostFaceNetV1-1	1.3	MS1MV3	99.7 ± 0.099	**95.9 ± 0.09**	90.4 ± 0.2	**96.8 ± 0.09**	**99.75 ± 0.3**	**97.9 ± 0.1**	**93 ± 0.1**	**94.9 ± 0.1**	**94.45 ± 0.09**
GhostFaceNetV1-2	1.3	MS1MV2	99.61 ± 0.1	95.5 ± 0.09	**90 ± 0.1**	91 ± 0.2	99.5 ± 0.213	96.15 ± 0.09	90.5 ± 0.09	92.6 ± 0.1722	91.4 ± 0.18
GhostFaceNetV1-2	1.3	MS1MV3	**99.68 ± 0.2494**	**95.55 ± 0.15**	88.556 ± 0.042	**93 ± 0.1**	**99.6 ± 0.129**	**96.9 ± 0.0501**	**91.18 ± 0.198**	**93.4 ± 0.1**	**92.05 ± 0.15**
GhostFaceNetV2-1	1.3	MS1MV2	99.8 ± 0.18	96 ± 0.258	92.9 ± 0.09	98.85 ± 0.13	99.8 ± 0.12	98.54 ± 0.1299	95.7 ± 0.1	96.95 ± 0.195	95.8 ± 0.27
GhostFaceNetV2-1	1.3	MS1MV3	**99.8 ± 0.2001**	**96.03 ± 0.1067**	**94.61 ± 0.12**	**99.28 ± 0.15**	**99.88 ± 0.1029**	**98.58 ± 0.12**	**96.445 ± 0.105**	**97.7 ± 0.15**	**96.1 ± 0.24**
GhostFaceNetV2-2	1.3	MS1MV2	**99.68 ± 0.11**	95.65 ± 0.15	89.54 ± 0.1299	93.03 ± 0.12	99.55 ± 0.1071	96.5 ± 0.15	91.7 ± 0.1872	93 ± 0.0972	92.2 ± 0.18
GhostFaceNetV2-2	1.3	MS1MV3	99.62 ± 0.189	**95.67 ± 0.189**	**90.1 ± 0.2001**	**94.22 ± 0.1971**	**99.6 ± 0.1299**	**96.77 ± 0.189**	**91.82 ± 0.207**	**93.1 ± 0.177**	**92.76 ± 0.18**

techniques previously proposed [53], [54]. In the ablation study, we conducted a set of experiments to determine the loss function that best fits the FR field and our GhostFaceNets in Section V-C.

V. EXPERIMENTAL SETUP

This section provides a comprehensive explanation of our training setup, the datasets we used, and the extensive ablation study that we performed to determine the optimal hyperparameters for our model and the FR/FV community.

A. DATASETS

We chose the MS1MV2 dataset, introduced in [9], and the MS1MV3 dataset, presented in [6], to train our models, based on prior works [9], [10], [11], [12], [20], [21], [22], [23]. Both MS1MV2 and MS1MV3 are cleaned versions of the MS-Celeb-1M dataset [55], and contain approximately 5.8 million faces of 85,000 identities and 5.1 million faces of 91,000 identities, respectively. We evaluate the performance of the GhostFaceNets models using these datasets in Section V-C.

During training, we continuously evaluate our model on FR/FV benchmarks by using the trained model as a feature extractor and computing the Cosine distance between feature vectors in all verification experiments. The test sets employed in this study cover a wide range of aspects and include LFW [28], AgeDB-30 [56], CFP-FP [57], CFP-FF [58], CP-LFW [58], CA-LFW [59], VGG2-FP [60], IJB-B [61], IJB-C [62], and MegaFace [63].

B. TRAINING SETUP

The proposed models in this paper are implemented using the Keras framework. Data preprocessing is performed using the Multi-Task Cascaded Convolutional Networks (MTCNN) solution [63] to detect and align face images. GhostFaceNets output a $512 - d$ embedding after processing input face images of size $112 \times 112 \times 3$. We used Stochastic Gradient Descent (SGD) optimizer with 0.9 momentum and cosine learning rate decay, starting at 0.1 and ending at 10^{-5}. The models were trained for 50 epochs with three different loss functions, namely ArcFace [9], CosFace [54], and Sub-center ArcFace [64]. We added l_2 regularization to the model's output layer with $l_2 = 1/2$ to prevent overfitting. Cosine distance is used for verification experiments. All experiments were performed using Python 3.9.13 and Keras on a workstation with a 1 Nvidia GeForce RTX 3080 GPU. Mixed precision [65] is used for faster training and less memory usage. The code will be made available upon publication.

The models in this study were trained using Keras mixed precision training [65], which uses lower precision (such as 16-bit) for certain parts of the model while maintaining acceptable performance using higher precision (such as 32-bit) for other parts of the model. This reduces memory usage and maintains numeric stability, resulting in faster computations and a reduced time spent transferring data between the CPU and GPU during training. The use of Automatic Mixed Precision (AMP) [65] dynamically adjusts the precision of computations to maintain numerical stability, allowing the model to use lower precision when possible and automatically switching to higher precision when necessary to avoid issues such as underflow or overflow. Incorporating mixed precision training in the training of GhostFaceNets on GPUs is motivated by practical considerations surrounding available hardware resources. By reducing memory requirements, mixed precision training allows for the efficient exploration of the potential landscape of model architectures and hyperparameters in numerous iterations and variations, as shown in Section V-C.

C. ABLATION STUDY - DETERMINING MODELS HYPERPARAMETERS

1) GhostFaceNetV1

Determining the width multiplier α is crucial for finding the optimal model balance between model complexity and performance. Therefore, five α values were tested and evaluated on the LFW, AgeDB-30, and CFP-FP testing datasets.

TABLE 7. The achieved results on 8 benchmarks. The results are reported in %. The models are ordered on the basis of the number of flops. Our GhostFaceNets-1 and GhostFaceNets-2 consistently extend the SOTA performance on all evaluation benchmarks for all models. All decimal points are provided as reported in the respective works. The best performance in each category on each benchmark is in bold, and ∗ indicates the best performance in all categories.

Method	#Params. (M)	FLOPs (M)	Model Size (MB)	LFW (%)	CA-LFW (%)	CP-LFW (%)	CFP-FP (%)	CFP-FF (%)	AgeDB _30 (%)	IJB-B (%)	IJB-C (%)
MobileFaceNetV1 [8]	3.4	1100	13.1	99.40	94.47	87.17	95.80	**99.50**	96.40	92.00	93.90
PocketNetM-256 [11]	1.75	1099.15	-	99.58	95.63	**90.03**	95.66	-	97.17	90.74	92.70
PocketNetM-128 [11]	1.68	1099.02	-	99.65	**95.67**	90.00	95.07	-	96.78	90.63	92.63
ShuffleFaceNet 2× [21]	4.5	1050	18	99.62	-	-	97.56	-	97.28	-	-
VarGFaceNet [22]	5	1022	20	**99.85**	95.15	88.55	**98.50**	**99.50**	**98.15**	**92.90**	**94.70**
AirFace [36]	-	1000	-	99.27	-	-	94.11	-	93.25	-	-
MobileFaceNet [8]	2	933.3	4	**99.70**	95.20	89.22	96.90	**99.60**	**97.60**	**92.80**	**94.70**
ProxylessFaceNAS [8]	3.2	900	12.5	99.20	92.55	84.17	94.70	98.80	94.40	87.10	89.70
MixFaceNet-M [23]	3.95	626.1	-	99.68	-	-	-	-	97.05	91.55	93.42
ShuffleMixFaceNet-M [23]	3.95	626.1	-	99.60	-	-	-	-	96.98	91.47	91.47
PocketNetS-256 [11]	0.99	587.224	3.9	99.66	**95.50**	**88.93**	93.34	-	96.35	89.31	91.33
PocketNetS-128 [11]	0.92	587.11	3.7	99.58	95.48	89.63	94.21	-	96.10	89.44	91.62
ShuffleFaceNet 1.5× [21]	2.6	577.5	10.5	99.67	95.05	88.50	**97.26**	-	97.32	92.30	94.30
MixFaceNet-S [23]	3.07	451.7	-	99.60	-	-	-	-	96.63	90.17	92.30
ShuffleMixFaceNet-S [23]	3.07	451.7	-	99.58	-	-	-	-	97.05	90.94	93.08
MobileFaceNets [20]	0.99	439.8	8.2	99.55	-	-	-	-	96.07	-	-
ShuffleFaceNet 1× [21]	1.4	275.8	5.6	99.45	-	-	96.04	-	96.33	-	-
GhostFaceNetV2-1 (MS1MV3) (ours)	6.88	272.105	13.743	**99.8667**∗	**96.1167**∗	**94.65**∗	**99.33**∗	**99.9143**∗	**98.62**∗	**96.48**∗	**97.75**∗
GhostFaceNetV2-1 (MS1MV2) (ours)	6.88	272.105	13.743	99.85	96.086	92.93	98.9143	99.84	98.5833	95.745	97.015
GhostFaceNetV1-1 (MS1MV3) (ours)	4.09	215.658	8.17	99.73	95.93	91.93	96.83	99.81	98	93.12	94.94
GhostFaceNetV1-1 (MS1MV2) (ours)	4.09	215.658	8.17	99.77	95.88	90.47	95.64	99.80	97.58	92.19	94.06
MixFaceNet-XS [23]	1.04	161.9	-	99.60	-	-	-	-	95.85	88.48	90.73
ShuffleMixFaceNet-XS [23]	1.04	161.9	-	99.53	-	-	-	-	95.62	87.86	90.43
GhostFaceNetV2-2 (MS1MV3) (ours)	6.84	76.513	13.663	99.683	**95.733**	**90.1667**	**94.2857**	**99.6443**	96.833	**91.889**	**93.159**
GhostFaceNetV2-2 (MS1MV2) (ours)	6.84	76.513	13.663	**99.7167**	95.7	89.5833	93.07	99.5857	96.55	91.7624	93.0324
ShuffleFaceNet 0.5× [21]	0.5	66.9	1.9	99.23	-	-	92.59	-	93.22	-	-
GhostFaceNetV1-2 (MS1MV3) (ours)	4.06	60.296	8.07	99.68	95.60	90.07	93.31	99.64	**96.92**	91.25	93.45
GhostFaceNetV1-2 (MS1MV2) (ours)	4.06	60.296	8.07	99.65	95.53	88.57	91.51	99.56	96.18	90.53	92.66

ArcFace loss function and a fixed stride of 2 were used in the experiment. Models with different α, their corresponding complexity, number of parameters, and accuracies on different benchmarks are presented in Table 1. It should be noted that all models in Table 1, 2, 4, 5, and 9 were trained on MS1MV2 dataset.

The effectiveness of various loss functions was further examined. In this experiment, three loss functions were

utilized, namely, ArcFace [9], CosFace [54], and Sub-center ArcFace [64], were utilized in this experiment. The width hyperparameter was fixed at $\alpha = 1.3$, as determined by the results of Table 1. The stride hyperparameter was set to $st = 1$ and $st = 2$. The models' loss functions and corresponding benchmark accuracy are presented in Table 2.

2) GhostFaceNetV2

A different sequence of experiments was conducted to examine the impact of these hyperparameters. The width multiplier hyperparameter α of the network was first determined by testing five α values and evaluating their performance in the LFW, AgeDB-30, and CFP-FP datasets. The ArcFace loss function was used, and the stride hyperparameter was fixed to 2 to accelerate training. Table 3 presents the various values of the models α, the corresponding complexity, the number of parameters, and the benchmark accuracy. In particular, these models were trained on both the MS1MV2 and MS1MV3 datasets, unlike Table 1.

Performance was again evaluated after altering the loss function. Similar to Table 2, in this experiment three loss functions were used, namely ArcFace [9], CosFace [54], and Sub-center ArcFace [64]. The width hyperparameter was fixed at $\alpha = 1.3$, as determined by the results of Table 3, and the stride hyperparameter was set to 2 to speed up training. The models were trained on the MS1MV2 dataset. Table 4 presents the models' loss functions and corresponding benchmark accuracy.

An experiment was also performed to identify the optimal recognition head settings. A conventional GDC head, commonly used in [20], [21], [22], and [23], was compared to our modified GDC. The width hyperparameter was fixed at $\alpha = 1.3$, as determined by the results of Table 3, the stride hyperparameter was set at 2 for faster training, and the ArcFace loss function was chosen based on the results of Table 4. Table 5 presents the model recognition heads and the corresponding benchmark accuracy.

3) GhostFaceNets

After fixing the width ($\alpha = 1.3$ based on the results of Table 1, and 3), the loss function (ArcFace based on Table 2, and 4 results), and the recognition head (Modified GDC based on Table 5), we used the two training datasets, namely MS1MV2 and MS1MV3, to train GhostFaceNets. We list down the performance comparison of our models with $st = 1$ and $st = 2$ in Table 6, along with their respective confidence intervals. No statistically significant differences were found between the models presented on the LFW, CFP-FF, and VGG2-FP benchmarks. However, significant improvements were observed in the CP-LFW, CFP-FP, AgeDB-30, and IJB-B benchmarks when using MS1MV3 instead of MS1MV2. The variation in performance on different testing datasets may be attributed to differences in the size, quality, and composition of the training dataset, as well as challenges such as variations in pose, illumination, and expression.

TABLE 8. Comparison of the practical inference time and model size of the GhostFaceNets.

Model	Inference Time (ms)	Model Size (MB)
GhostFaceNet V1-1	46	8.17
GhostFaceNet V1-2	22	8.07
GhostFaceNet V2-1	52.4	13.706
GhostFaceNet V2-2	28.4	13.606

TABLE 9. The performance of the proposed GhostFaceNets with width $\alpha = 2$.

Model	α	Com-plexity FLOPS	# Params.	LFW (%)	AgeDB _30 (%)	CFP_ FP (%)
GhostFaceNet V1-1	2	480,317, 128	9,222,1 44	99.77	97.47	96.56
GhostFaceNet V1-1	2	613,994, 208	15,412, 944	99.85	98.2	98.53

Based on the ablation study, eight models were adopted. These models have a stride of 1 and 2, a width of $\alpha = 1.3$, a modified GDC as recognition head, and use the ArcFace loss function. These models were trained on the MS1MV2 and MS1MV3 datasets. The notation GhostFaceNetVi-st (MS1MVk) represents GhostFaceNetVi of the stride st trained on MS1MVk, where $i \in \{1, 2\}$, $st \in \{1, 2\}$ and $k \in \{1, 2\}$. The eight models are as follows: GhostFaceNetV1-1 (MS1MV2), GhostFaceNetV1-2 (MS1MV2), GhostFaceNetV1-1 (MS1MV3), GhostFaceNetV1-2 (MS1MV3), GhostFaceNetV2-1 (MS1MV2), GhostFac-eNetV2-2 (MS1MV2), GhostFaceNetV2-1 (MS1MV3), and GhostFaceNetV2-2 (MS1MV3).

We also used the TFLite tool [65] to gauge the real inference speed of the proposed GhostFaceNets on an ARM-based mobile phone because they are intended for mobile applications. Following the common settings in [24] and [25], we use a single-threaded mode with batch size 1. Table 8 shows the inference time and model size of our proposed GhostFaceNets.

As a side experiment (which will not be considered in the selection of hyperparameters for our models), we designed large GhostFaceNets-1 of width $\alpha = 2$ to compare it with large FR models. We trained the model on MS1MV2 utilizing ArcFace as the loss function. The results of the analysis are presented in Table 9, which demonstrates that even with an increase in the size of the model, the performance remains relatively consistent.

VI. RESULTS

This section presents the results of GhostFaceNets models on various benchmarks and compares them with previous studies. While adhering to the evaluation methodologies and criteria of each benchmark and previous research, recognition

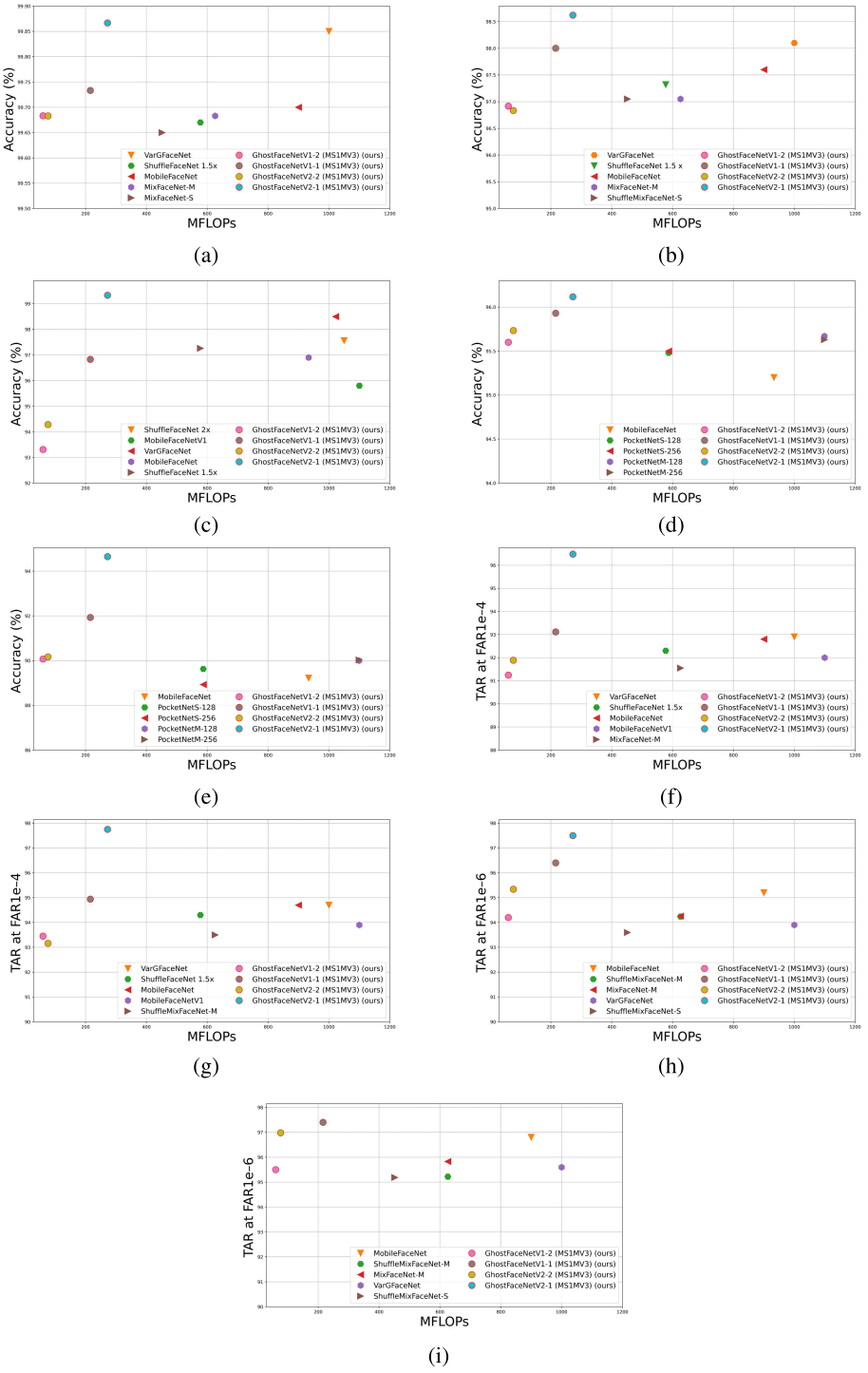

FIGURE 6. Number of FLOPs (in millions) vs. performance on (a) LFW (accuracy), (b) AgeDB-30 (accuracy), (c) CFP-FP (accuracy), (d) CA-LFW (accuracy), (e) CP-LFW (accuracy), (f) IJB-B (TAR at FAR1e-4), (g) IJB-C (TAR at FAR1e-4), (h) MegaFace (TAR at FAR1e-6), and (i) MegaFace(R) (TAR at FAR1e-6). Our GhostFaceNets are marked with a circle marker and red edge color and are placed repeatedly in the top left corner, proving a SOTA trade-off between FR performance and computational complexity.

and verification performance are evaluated according to ISO/IEC 19795-1 [66] to enhance reproducibility and comparability.

Table 7 presents the FR/FV results of our proposed GhostFaceNets models, comparing them to recent SOTA compact models on nine benchmarks. The models are organized

TABLE 10. The achieved results on MegaFace and refined MegaFace(R) Challenge1 using FaceScrub as the probe set. "Rank-1" refers to the accuracy of Rank-1 face identification in reported % with 1M distractors, and "Ver." refers to the face verification given in TAR at 10^{-6} FAR. All decimal points are provided as reported in the respective works. The best performance is in bold.

Method	Param. (M)	FLOPs (M)	Model Size (MB)	MegaFace		MegaFace(R)	
				Rank-1 (%)	Ver. (%)	Rank-1 (%)	Ver. (%)
ShuffleMixFaceNet-S	3.07	451.7	-	77.41	93.60	94.07	95.19
ShuffleMixFaceNet-M	3.95	626.1	-	78.13	94.24	94.64	95.22
MixFaceNet-M	3.95	626.1	-	78.2	94.26	94.95	95.83
MobileFaceNet	2	933	4	79.3	95.2	95.8	96.8
VarGFaceNet	5	1022	20	78.2	93.9	94.9	95.6
GhostFaceNetV1-2 (MS1MV2)	4.06	60.296	8.07	78.24	94.01	94.98	95.86
GhostFaceNetV1-2 (MS1MV3)	4.06	60.296	8.07	78.3	94.2	95.1	95.5
GhostFaceNetV2-2 (MS1MV2)	6.84	76.513	13.663	79.31	95.21	95.83	96.86
GhostFaceNetV2-2 (MS1MV3)	6.84	76.513	13.663	79.35	95.34	96.01	96.98
GhostFaceNetV1-1 (MS1MV2)	4.09	215.658	8.17	79.32	96.2	95.94	96.9
GhostFaceNetV1-1 (MS1MV3)	4.09	215.658	8.17	79.4564	96.4	96.3791	97.4
GhostFaceNetV2-1 (MS1MV2)	6.88	272.105	13.743	80.795	96.77	98.22	98.31
GhostFaceNetV2-1 (MS1MV3)	6.88	272.105	13.743	**83.2**	**97.5**	**98.64**	**98.72**

into four groups based on complexity (FLOPs): above 1000, 500-1000, 100-500, and below 100 MFLOPs.

GhostFaceNetsV1 trained on the MS1MV3 dataset achieved SOTA performance on most benchmarks. Starting with the stride of 1. GhostFaceNetV1-1 (MS1MV3) outperformed all lightweight SOTA models in groups 1-4 on 8 benchmarks, except for the LFW, AgeDB-30, and CFP-FP benchmarks. It achieved 99.73% and 98% on LFW and AgeDB-30, respectively, only slightly behind VarGFaceNet. On CFP-FP, it achieved 96.83% accuracy, behind models with much higher FLOPs, such as VarGFaceNet with 1022 MFLOPs, ShuffleFaceNet 1.5× with 577.5 MFLOPs, and ShuffleFaceNet 2× with 1050 MFLOPs. Notably, GhostFaceNetV1-1 has only 215.658 MFLOPs, significantly lower than the models that slightly outperformed it.

For stride 2, GhostFaceNetV1-2 (MS1MV3) outperformed lightweight SOTA models in groups 3 and 4 on all benchmarks, except for a 0.1% difference on AgeDB-30 against ShuffleMixFaceNet-S. On CP-LFW and CFP-FF, it outperformed all SOTA models in groups 1 and 2. It achieved a competitive accuracy on LFW, beating all SOTA models except VarGFaceNet and MobileFaceNet. On CA-LFW, it achieved 95.6% accuracy, ranking third behind PocketNetM-128 and PocketNetM-256. On CFP-FP and AgeDB-30, it achieved 93.3143% and 96.9167% accuracy, respectively, compared to the best verification accuracy of 98.5% and 98.15% achieved by VarGFaceNet. Notably, GhostFaceNetV1-2 has only 60.296 MFLOPs, which is even lower than GhostFaceNetV1-1's FLOPs.

GhostFaceNets trained on the MS1MV2 dataset achieved relatively lower performance compared to GhostFaceNets trained on MS1MV3. GhostFaceNetV1-1 (MS1MV2) and GhostFaceNetV1-2 (MS1MV2) outperformed groups 3 and 4 on all benchmarks, and achieved competitive results against groups 1 and 2, even outperforming them in some cases. For example, GhostFaceNetV1-1 (MS1MV2) achieved SOTA accuracy on CFP-FF, CA-LFW, and CP-LFW among all models in groups 1 and 2, and GhostFaceNetV1-2 (MS1MV2) achieved SOTA accuracy on CFP-FF among all models in groups 1 and 2.

On the large-scale evaluation benchmarks IJB-B and IJB-C, GhostFaceNetV1-1 (MS1MV3) achieved SOTA performance with 93.116% and 94.943% TAR at FAR 10^{-4}, respectively. The other GhostFaceNets also achieved competitive results on IJB-B and IVB-C, such as GhostFaceNetV1-2 (MS1MV3) achieving 91.246% and 93.45%, GhostFaceNetV1-1 (MS1MV2) achieving 92.191% and

TABLE 11. Ghost bottlenecksV1 hyperparameters. *exp* means expansion size, *out* means the number of output channels, *d* is the filter size in each Ghost bottleneck, *exp[new]* is the modified expansion size, *out[new]* is the modified number of output channels, and *pre-out* is the previous number of output channels.

i	*d*	*exp*	*out*	*exp[new]*	*out[new]*	*pre-out*
0	-	-	16	-	-	-
-	-	-	-	-	-	-
1	3	16	16	20	20	20
2	3	48	24	64	32	20
3	3	72	24	92	32	32
4	5	72	40	922	52	32
5	5	120	40	156	52	52
6	3	240	80	312	104	52
7	3	200	80	260	104	104
8	3	184	80	240	104	104
9	3	184	80	240	104	104
10	3	480	112	624	144	104
11	3	672	112	872	144	144
12	5	672	160	872	208	144
13	5	960	160	1248	208	208
14	5	960	160	1248	208	208
15	5	960	160	1248	208	208
16	5	960	160	664	208	208
-	-	-	960	-	-	-
-	-	-	-	-	-	-

94.058%, GhostFaceNetV1-2 (MS1MV2) achieving 90.526% and 92.6574%. The best verification performance on these benchmarks is 92.8% and 94.7% achieved by MobileFaceNet, and 92.9% and 94.7% achieved by VarGFaceNet, respectively.

Our GhostFaceNetsV2 models demonstrated even more superior performance, particularly when using a stride of 1. GhostFaceNetV2-1 (MS1MV3) achieved SOTA performance on all benchmarks, outperforming all previous FR models in the literature. GhostFaceNetV2-1 (MS1MV2) also surpassed all FR models in the literature, though not as well as GhostFaceNetV2-1 (MS1MV3). This shows that GhostFaceNetsV2 models are highly effective for FR tasks and represent a significant improvement over GhostFaceNetsV1 models. A similar trend of improved accuracy is observed when comparing GhostFaceNetsV2-2 with GhostFaceNetsV1-2. However, it should be noted that GhostFaceNetsV2-2 still performed lower than some models with high computational complexity (above 1000 MFLOPs) on certain benchmarks, such as VarGFaceNet on LFW (99.85% vs 99.72%). This suggests that the GhostFaceNetsV2-2 models demonstrate an improvement in performance over GhostFaceNetsV1-2 models and are able to achieve results that are comparable to or better than models with higher computational complexity on all benchmarks.

Furthermore, our study evaluated the performance of GhostFaceNets on MegaFace and its refined version (R), comparing them to existing models in the literature. We chose to show the latest top 5 SOTA models shown in Table 10. Based on the results presented in Table 10, it is evident that GhostFaceNet outperforms the other methods on both MegaFace and MegaFace(R) datasets. Specifically, GhostFaceNetV2-1 achieves the highest Rank-1 face identification accuracy of 83.2% and the highest verification rate of 97.5% at 10^{-6} FAR on MegaFace and achieves the highest Rank-1 accuracy of 98.64% and verification rate of 98.72% at 10^{-6} FAR on MegaFace(R). These results demonstrate that GhostFaceNetV2-1 is superior to other SOTA models in terms of face recognition performance on large-scale datasets, which is a crucial task for various real-world applications.

To visually demonstrate the efficiency of our GhostFaceNets, we compared the number of FLOPs with the verification performance achieved in Tables 7 and 10 in Figure 6, for our GhostFaceNets and the top five compact models that perform best in the recent literature in each benchmark. We chose the most effective GhostFaceNets, trained on the MS1MV3 dataset, to convey the capabilities of GhostFaceNets more effectively. Figure 6 shows that our GhostFaceNets consistently appear in the upper left corner, indicating a superior trade-off between model complexity and FR performance compared to other approaches.

VII. CONCLUSION

In this paper, we introduced GhostFaceNets, highly accurate and effective facial recognition models. Many experiments were conducted on well-known publicly available datasets including LFW, AgeDB-30, and large-scale datasets such as IJB-B, IJB-C, and MegaFace. The findings of the overall study show that our proposed GhostFaceNets are effective for applications with minimal computational complexity constraints. It has been found that among a range of models with varying computational complexity, from 0 MFLOPs to 1000 MFLOPs, the GhostFaceNets have demonstrated exceptional performance on all the benchmarks used to evaluate their capabilities. In particular, they have achieved SOTA performance, indicating that they are among the most advanced models in their field in terms of both efficiency and

TABLE 12. Proposed GhostFaceNet*Vi*-1 architecture.

i	Input Shape (V1)	Operator (V1)	Input Shape (V2)	Operator (V2)	SE ratio r	Stride	Shortcut
0	$112^2 \times 3$	Conv2d 3×3	$112^2 \times 3$	Conv2d 3×3	-	1	-
-	$112^2 \times 20$	BN + PReLU	$112^2 \times 20$	BN+ PReLU	-	-	-
1	$112^2 \times 20$	G-bneck V1	$112^2 \times 20$	G-bneck V1	-	1	False
2	$112^2 \times 20$	G-bneck V1	$112^2 \times 20$	G-bneck V1	-	2	True
3	$56^2 \times 32$	G-bneck V1	$56^2 \times 32$	G-bneck V2	-	1	False
4	$56^2 \times 32$	G-bneck V1	$56^2 \times 32$	G-bneck V2	0.25	2	True
5	$28^2 \times 52$	G-bneck V1	$28^2 \times 52$	G-bneck V2	0.25	1	False
6	$28^2 \times 52$	G-bneck V1	$28^2 \times 52$	G-bneck V2	-	2	True
7	$14^2 \times 104$	G-bneck V1	$14^2 \times 104$	G-bneck V2	-	1	False
8	$14^2 \times 104$	G-bneck V1	$14^2 \times 104$	G-bneck V2	-	1	False
9	$14^2 \times 104$	G-bneck V1	$14^2 \times 104$	G-bneck V2	-	1	False
10	$14^2 \times 104$	G-bneck V1	$14^2 \times 104$	G-bneck V2	0.25	1	True
11	$14^2 \times 104$	G-bneck V1	$14^2 \times 104$	G-bneck V2	0.25	1	False
12	$14^2 \times 104$	G-bneck V1	$14^2 \times 104$	G-bneck V2	0.25	2	True
13	$7^2 \times 208$	G-bneck V1	$7^2 \times 208$	G-bneck V2	-	1	False
14	$7^2 \times 208$	G-bneck V1	$7^2 \times 208$	G-bneck V2	0.25	1	False
15	$7^2 \times 208$	G-bneck V1	$7^2 \times 208$	G-bneck V2	-	1	False
16	$7^2 \times 208$	G-bneck V1	$7^2 \times 208$	G-bneck V2	0.25	1	False
-	$7^2 \times 208$	Conv2d 1×1	$7^2 \times 208$	Conv2d 1×1	-	1	-
-	$7^2 \times 664$	BN + PReLU	$7^2 \times 1248$	BN + PReLU	-	-	-
-	$7^2 \times 664$	DW Conv 7×7	$7^2 \times 1248$	DW Conv 7×7	-	1	-
-	$1^2 \times 664$	BN	$1^2 \times 1248$	BN	-	-	-
-	$1^2 \times 664$	Conv2d 1×1	$1^2 \times 1248$	Conv2d 1×1	-	1	-
-	$1^2 \times 512$	Flatten	$1^2 \times 512$	Flatten	-	-	-
-	None × 664	BN + Linear Activation	None × 512	BN + Linear Activation	-	-	-

TABLE 13. Comparison between large SOTA FR models and our GhostFaceNets. The best performance in each benchmark is in bold.

Method	Param. (M)	FLOPs (M)	LFW (%)	AgeDB_30 (%)	CFP-FP (%)	CFP-FF (%)	CP-LFW (%)	CA-LFW (%)	IJB-B (%)	IJB-C (%)	MegaFace		MegaFace(R)	
											Rank-1 (%)	Ver. (%)	Rank-1 (%)	Ver. (%)
FaceNet	3.07	451.7	99.63	-	-	-	-	-	-	-	70.49	86.47	-	-
SphereFace	65.2	24211	99.42	-	-	-	81.40	90.30	-	-	72.729	85.561	-	-
CosFace	65.2	24211	99.73	-	-	-	-	-	-	-	80.56	96.56	97.91	97.91
ArcFace	65.2	24211	99.83	98.15	98.37	-	92.08	95.45	94.2	95.65	81.03	96.98	98.35	98.48
SFace	65.2	24211	99.82	95.10	95.81	-	93.28	96.07	-	96.11	-	-	98.50	98.61
Prodploy	65.2	24211	99.83	98.467	98.986	99.886	93.317	**96.233**	95.19	96.58	-	-	**98.78**	**98.95**
GhostFaceNetV1-2 (MS1MV2)	4.06	60.296	99.65	96.18	91.51	99.5571	90.0667	95.53	90.5258	92.6574	78.24	94.01	94.98	95.86
GhostFaceNetV1-2 (MS1MV3)	4.06	60.296	99.65	96.9167	93.31	99.643	88.57	95.6	91.246	93.45	78.3	94.2	95.1	95.5
GhostFaceNetV2-2 (MS1MV2)	6.84	76.513	99.7167	96.55	93.07	99.5857	89.5833	95.7	91.7624	93.0324	79.31	95.21	95.83	96.86
GhostFaceNetV2-2 (MS1MV3)	6.84	76.513	99.683	96.833	94.2857	99.6443	90.2857	95.733	91.889	93.159	79.35	95.34	96.01	96.98
GhostFaceNetV1-1 (MS1MV2)	4.09	215.658	99.7667	97.58	95.64	99.8	91.9333	95.88	92.191	94.058	79.32	96.2	95.94	96.9
GhostFaceNetV1-1 (MS1MV3)	4.09	215.658	99.7333	98	96.83	99.81	90.4667	95.93	93.116	94.943	79.4564	96.4	96.3791	97.4
GhostFaceNetV2-1 (MS1MV2)	6.88	272.105	99.85	98.5833	98.9143	99.84	92.93	96.086	95.745	97.015	80.795	96.77	98.22	98.31
GhostFaceNetV2-1 (MS1MV3)	6.88	272.105	**99.8667**	**98.62**	**99.33**	**99.9143**	**94.65**	96.1167	**96.48**	**97.75**	**83.2**	**97.5**	98.64	98.72

effectiveness. This is a significant achievement, as it suggests that the GhostFaceNets can deliver superior results while also being efficient in their use of computational resources.

APPENDIX

A. GhostNets BOTTLENECK

At each stage of the stacked Ghost bottlenecksV1, the filter sizes d, the expansion size exp, the number of output channels out, and the previous number of output channels are summarized in Table 11. The exp and out are obtained from [24]. The new expansion size *exp[new]*, and the new number of output channels *out[new]* are computed using,

$$\begin{aligned} out[new] &= \_make\_divisible(exp[i]\times\alpha, 4)_i \in [1, 2, \ldots, 16], \\ exp[new] &= \_make\_divisible(exp[i]\times\alpha, 4)_i \in [1, 2, \ldots, 16], \end{aligned} \tag{2}$$

where the width multiplier is $\alpha = 1.3$. make_divisible function ensures that all layers have a channel number that is divisible by 8. And the previous number of output channels is basically given by,

$$\begin{aligned} pre - out[i] &= out[new][i-1] i \in [2, 3, \ldots, 16], \\ pre - out[1] &= 20, \end{aligned} \tag{3}$$

The Ghost bottlenecksV1' hyperparameters are displayed in Table 11. The hyperparameters described below are for the stride of $st = 1$ Ghost bottleneckV1. Regarding the stride of $st = 2$ Ghost bottleneckV1, a depthwise convolution with a stride of (st = 2) is placed between two Ghost modules, as shown in Figure 3 (c) and (d).

The hyperparameters Ghost bottleneckV2 are kept the same as in Table 11 with a small modification of using the parallel DFC attention branch and Ghost module branch from bottleneck $i = 3$ till bottleneck $i = 16$.

B. GhostFaceNets ARCHITECTURE

The detailed architecture of the proposed GhostFaceNetVi-1 $i \in \{1, 2\}$ is shown in Table 12 and Figure 4. In Table 12, the BN denotes Batch Normalization [43], G-bneck Vi denotes GhostVi bottleneck, and the Shortcut term denotes that the concept of shortcut was used in the network to connect the input to the output when $out[new][i-1] = pre - out[i]$ & $strides > 1$. This is used to avoid degradation issue and preserve information. The shortcut is basically a depthwise convolution of $d \times d$ kernel size and stride of $st = 2$ followed by batch normalization [43], convolution, and again batch normalization [43] as shown in Figure 2 (e).

C. GhostFaceNet COMPARISON WITH OTHER SOTA LARGE FACE RECOGNITION MODELS

In order to demonstrate the superiority of our proposed model for FR, we conducted a comprehensive comparative study with SOTA large models in the literature. This comparison was made on nine benchmarks, including LFW [28], AgeDB [56], and large-scale datasets, such as MegaFace [63], its refined version (R), and the benchmarks IJB-B [61] and IJB-C [62]. The main objective of this study was to evaluate the performance of our model against existing SOTA approaches and highlight its strengths and weaknesses compared to the current SOTA.

The comparative study involved a detailed evaluation of various FR models, including ArcFace [9], ElasticFace [10], FaceNet [32], SphereFace [53], CosFace [54], Prodpoly [68], SFace [69], and our proposed GhostFaceNets. The evaluation was based on accuracy, TAR at 10^{-4} FAR, and TAR at 10^{-6} FAR.

Our approach outperformed all other models and achieved SOTA results on all benchmarks except CA-LFW [59] and the refined version of MegaFace [63] by a significant margin. On CA-LFW and MegaFace refined, our approach achieved an accuracy of 96.1167% and a Rank-1 accuracy of 98.64%, and a verification TAR of 98.72% at a FAR of 10^{-6} compared to the current SOTA model, Prodpoly, which achieved an accuracy of 96.233%, a Rank-1 accuracy of 98.78%, and a verification TAR of 98.95%.

Overall, the results of the comparative study demonstrate the effectiveness of our proposed model and its superiority over the existing SOTA models for FR tasks.

REFERENCES

[1] W. Ahmed, A. Rasool, A. R. Javed, N. Kumar, T. R. Gadekallu, Z. Jalil, and N. Kryvinska, "Security in next generation mobile payment systems: A comprehensive survey," *IEEE Access*, vol. 9, pp. 115932–115950, 2021.

[2] E. Vazquez-Fernandez and D. Gonzalez-Jimenez, "Face recognition for authentication on mobile devices," *Image Vis. Comput.*, vol. 55, pp. 31–33, Nov. 2016.

[3] C. Morikawa, M. Kobayashi, M. Satoh, Y. Kuroda, T. Inomata, H. Matsuo, T. Miura, and M. Hilaga, "Image and video processing on mobile devices: A survey," *Vis. Comput.*, vol. 37, no. 12, pp. 2931–2949, Dec. 2021.

[4] R. K. Sinha, R. Pandey, and R. Pattnaik, "Deep learning for computer vision tasks: A review," 2018, *arXiv:1804.03928*.

[5] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, Mar. 2021.

[6] J. Deng, J. Guo, D. Zhang, Y. Deng, X. Lu, and S. Shi, "Lightweight face recognition challenge," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2638–2646.

[7] Y. Deng, "Deep learning on mobile devices: A review," in *Proc. SPIE*, vol. 10993, pp. 52–66, May 2019.

[8] 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," *Artif. Intell. Rev.*, vol. 54, pp. 6201–6244, Feb. 2021.

[9] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4690–4699.

[10] F. Boutros, N. Damer, F. Kirchbuchner, and A. Kuijper, "ElasticFace: Elastic margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 1578–1587.

[11] 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," *IEEE Access*, vol. 10, pp. 46823–46833, 2022.

[12] F. Boutros, N. Damer, and A. Kuijper, "QuantFace: Towards lightweight face recognition by synthetic data low-bit quantization," in *Proc. 26th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2022, pp. 855–862.

[13] M. Liadis, H. Wang, R. Molina, and A. K. Katsaggelos, "Robust and low-rank representation for fast face identification with occlusions," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2203–2218, May 2017.

[14] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size," 2016, *arXiv:1602.07360*.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[16] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[17] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.

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

[19] M. Tan and Q. V. Le, "MixConv: Mixed depthwise convolutional kernels," 2019, *arXiv:1907.09595*.

[20] S. Chen, Y. Liu, X. Gao, and Z. Han, "MobileFaceNets: Efficient CNNs for accurate real-time face verification on mobile devices," in *Biometric Recognition* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2018, pp. 428–438.

[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 *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2–4.

[22] 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 *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2–5.

[23] F. Boutros, N. Damer, M. Fang, F. Kirchbuchner, and A. Kuijper, "MixFaceNets: Extremely efficient face recognition networks," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Aug. 2021, pp. 1–8.

[24] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1580–1589.

[25] Y. Tang, K. Han, J. Guo, C. Xu, C. Xu, and Y. Wang, "GhostNetV2: Enhance cheap operation with long-range attention," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 1–12.

[26] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. E. Hinton, "On rectified linear units for speech processing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 3517–3521.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[28] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," in *Proc. Workshop Faces Real-Life Images, Detection, Alignment, Recognit.*, 2008, pp. 5–10.

[29] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua, "Labeled faces in the wild: A survey," in *Proc. Adv. Face Detection Facial Image Anal.*, 2016, pp. 189–248.

[30] X. Wu, R. He, Z. Sun, and T. Tan, "A light CNN for deep face representation with noisy labels," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2884–2896, Nov. 2018.

[31] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9127–9135.

[32] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[34] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.

[35] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, pp. 1789–1819, Mar. 2021.

[36] X. Li, F. Wang, Q. Hu, and C. Leng, "AirFace: Lightweight and efficient model for face recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, Oct. 2019, pp. 3–4.

[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[38] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 1997–2017, 2019.

[39] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," 2018, *arXiv:1806.09055*.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[42] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[43] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[44] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[46] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 15908–15919.
[47] K. Han, J. Guo, Y. Tang, and Y. Wang, "PyramidTNT: Improved transformer-in-transformer baselines with pyramid architecture," 2022, *arXiv:2201.00978*.
[48] Y. Tang, K. Han, C. Xu, A. Xiao, Y. Deng, C. Xu, and Y. Wang, "Augmented shortcuts for vision transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 15316–15327.
[49] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted Windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.
[50] S. Mehta and M. Rastegari, "MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer," 2021, *arXiv:2110.02178*.
[51] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, Atlanta, GA, USA, 2013, p. 3.
[52] M. Khalid, J. Baber, M. K. Kasi, M. Bakhtyar, V. Devi, and N. Sheikh, "Empirical evaluation of activation functions in deep convolution neural network for facial expression recognition," in *Proc. 43rd Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2020, pp. 204–207.
[53] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 212–220.
[54] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5265–5274.
[55] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-Celeb-1M: A dataset and benchmark for large-scale face recognition," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), Amsterdam, The Netherlands: Springer, 2016, pp. 87–102.
[56] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou, "AgeDB: The first manually collected, in-the-wild age database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 51–59.
[57] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs, "Frontal to profile face verification in the wild," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–9.
[58] T. Zheng and W. Deng, "Cross-pose LFW: A database for studying cross-pose face recognition in unconstrained environments," Beijing Univ. Posts Telecommun., Beijing, China, Tech. Rep., 2018, doi: 10.48550/arXiv.1708.08197.
[59] T. Zheng, W. Deng, and J. Hu, "Cross-age LFW: A database for studying cross-age face recognition in unconstrained environments," 2017, *arXiv:1708.08197*.
[60] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age," in *Proc. 13th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2018, pp. 67–74.
[61] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen, J. Cheney, and P. Grother, "IARPA Janus Benchmark-B face dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 90–98.
[62] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, and P. Grother, "IARPA Janus Benchmark—C: Face dataset and protocol," in *Proc. Int. Conf. Biometrics (ICB)*, Feb. 2018, pp. 158–165.
[63] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The MegaFace benchmark: 1 million faces for recognition at scale," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4873–4882.
[64] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.
[65] J. Deng, J. Guo, T. Liu, M. Gong, and S. Zafeiriou, "Sub-center ArcFace: Boosting face recognition by large-scale noisy web faces," in *Computer Vision—ECCV* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2020, pp. 741–757.
[66] M. Abadi, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2015, *arXiv:1603.04467*.
[67] *Infor-Mation Technology—Biometric Performance Testing and Reporting—Part 1: Principles and Framework*, Standard ISO/IEC 19795-1:2021, 2021.
[68] G. G. Chrysos, S. Moschoglou, G. Bouritsas, J. Deng, Y. Panagakis, and S. Zafeiriou, "Deep polynomial neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4021–4034, 2021.
[69] Y. Zhong, W. Deng, J. Hu, D. Zhao, X. Li, and D. Wen, "SFace: Sigmoid-constrained hypersphere loss for robust face recognition," *IEEE Trans. Image Process.*, vol. 30, pp. 2587–2598, 2021.

MOHAMAD ALANSARI (Member, IEEE) received the B.Sc. degree in electrical and computer engineering from Khalifa University, where he is currently pursuing the M.Sc. degree in electrical and computer engineering. His research interest includes computer vision in autonomous robotics.

OUSSAMA ABDUL HAY received the B.Sc. degree in mechanical engineering from the American University of Sharjah, United Arab Emirates, in 2015, and the M.Sc. degree in mechanical engineering from The University of Manchester, U.K., in 2016. He is currently pursuing the Ph.D. degree in robotics with the Center for Autonomous Robotic Systems, Khalifa University, Abu Dhabi, United Arab Emirates. His research interests include perception for navigation, visual servoing, and the applications of AI in robotics.

SAJID JAVED received the B.Sc. degree in computer science from the University of Hertfordshire, U.K., in 2010, and the master's and Ph.D. degrees in computer science from Kyungpook National University, Republic of Korea, in 2017. He is currently an Assistant Professor of computer vision with the Department of Electrical and Computer Engineering (ECE), Khalifa University, United Arab Emirates. Prior to that, he was a Research Scientist with the Khalifa University Center for Autonomous Robotics System (KUCARS), from 2019 to 2021. Before joining Khalifa University, he was a Research Fellow with the University of Warwick, U.K., from 2017 to 2018, where he worked on histopathological landscapes for better cancer grading and prognostication. His research interests include visual object tracking in the wild, multi-object tracking, background-foreground modeling from video sequences, moving object detection from complex scenes, and cancer image analytics, including tissue phenotyping, nucleus detection, and nucleus classification problems. His research themes involve developing deep neural networks, subspace learning models, and graph neural networks.

ABDULHADI SHOUFAN received the Dr.-Ing. degree from Technische Universität Darmstadt, Germany, in 2007. He is currently an Associate Professor of electrical engineering and computer science with Khalifa University, Abu Dhabi. His research interests include drone security and safe operation, embedded security, cryptography hardware, learning analytics, and engineering education.

YAHYA ZWEIRI (Member, IEEE) received the Ph.D. degree from King's College London, in 2003. He is currently an Associate Professor with the Department of Aerospace Engineering and the Deputy Director of the Advanced Research and Innovation Center, Khalifa University, United Arab Emirates. He was involved in defense and security research projects in the last 20 years at the Defense Science and Technology Laboratory, King's College London, and the King Abdullah II Design and Development Bureau, Jordan. He has published over 130 refereed journal articles and conference papers and has filed ten patents in the USA and the U.K. His main research interest includes robotic systems for extreme conditions with a particular emphasis on applied AI aspects and neuromorphic vision systems.

NAOUFEL WERGHI (Senior Member, IEEE) received the Ph.D. degree in computer vision from the University of Strasbourg, Strasbourg, France, in 1996. He was a Research Fellow with the Division of Informatics, The University of Edinburgh, Edinburgh, U.K., and a Lecturer with the Department of Computer Sciences, University of Glasgow, Glasgow, U.K. He was a Visiting Professor with the Department of Electrical and Computer Engineering, University of Louisville, Louisville, KY, USA. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Khalifa University, Abu Dhabi, United Arab Emirates. His research interests include image analysis and interpretation, where he has been leading several funded projects in the areas of biometrics, medical imaging, and intelligent systems.