BinaryFaceNet: A Binarized Approach for Real-Time Very Low Resolution Face Recognition in Video Surveillance Scenarios

Luis S. Luevano[1] Leonardo Chang[1] Miguel González-Mendoza[1] Yoanna Martínez-Díaz[2]
Heydi Méndez-Vázquez[2] Gilberto Ochoa Ruiz[1]
[1]Tecnologico de Monterrey
[2]Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV)
{luis.s.luevano, lchang, mgonza, gilberto.ochoa}@tec.mx
{ymartinez, hmendez}@cenatav.co.cu

Abstract

With the rise of Deep Learning methods in the past decade, we have seen a tremendous leap in accuracy performance for various computer vision tasks, such as face recognition. Training and inference for these methods often require expensive hardware equipment. In more recent years, efficient Deep Learning-based methods have emerged, with Lightweight and Quantized Convolutional Neural Networks being useful in computer vision applications for constrained hardware platforms. However, most approaches have been proposed for general-purpose tasks in mind. In this paper, we propose BinaryFaceNet, for very low resolution face recognition in the wild using an original Quantized Convolutional Neural Network design. We reduce the overhead of operations when downsizing feature maps with grouped convolutions, using a progressive downsizing in the Squeeze-and-Excitation block for a more accurate weighted signal and novel Global Feature Extraction and Local Feature Extraction blocks with DepthWise convolutions and alternated residual connections to avoid signal degradation due to quantization and network depth. With our implementation, we reduce Multiply-Accumulate operations in half with respect to the state of the art in Quantized Convolutional Neural Networks, while achieving a limited compromise in recognition performance in Very Low Resolution benchmark datasets.

1. Introduction

Deep Learning has allowed for more precise Machine Learning implementations for many tasks. In Computer Vision, Deep Convolutional Neural Networks (CNNs) have many applications such as image segmentation [26], character recognition [25], face recognition, among others. Surveillance applications with face identification and verification main challenges are uncontrolled conditions, bad image quality, and occlusion, lack of dataset availability for real-world data and the lack of mainstream research for real-time applications [19].

In recent years, the need of CNNs for solving Computer Vision problems using hardware-constrained devices has emerged. One approach to tackle such problem is using Lightweight CNNs networks such as SqueezeNet [16], ShuffleNetV2 [20], and MobileNets [14, 15, 28]. These lightweight CNNs have lowered thresholds for Floating Point Operations (FLOPS) to 1.2GFLOPS and memory footprint up to a model size of 20MB, much less compared to regular CNNs [9]. Some of the most successful methods tested for Lightweight High Resolution (HR) face recognition are ShuffleFaceNet [23], MobileFaceNet [3], and VarGFaceNet [32]. Tailored methods for Very Low Resolution (VLR) face recognition include [34], [18], and [35]. However, these are not lightweight approaches and only generalize to certain datasets. This makes them not suitable for a more general open-set identification real-world application scenarios. This effectively makes generalization a big gap in the state of the art. A study of Lightweight VLR face recognition for surveillance scenarios was carried out by Martínez-Díaz et al. [24], showing good results when bridging the resolution gap using bicubic and inter-area interpolation to simulate VLR conditions.

A more efficient approach is presented by Quantized and Binary Neural Networks (BNNs), using k-bits to approximate floating-point weights. In turn, binarized layer convolutions are more efficient using bit-shift, bit-count, and XNOR operations [2, 6, 7, 37]. Furthermore, various methods for optimizing binary weights have been proposed such as in [13] and [29]. Experimentally, some BNNs can speedup a real-value layers up to 58%, against a theoretical 62.27% [27].

The most important challenge to solve using binary net-

works is the loss of information when quantizing the intermediate representations. In state-of-the-art Lighweight CNNs, the signals are preserved by avoiding pooling representations, using larger strided convolutions, and using point-wise convolutions. However, this is mostly not suitable for binary networks because most of the time the goal is to quantize weights and layers to a binary representation. To achieve this quantization, authors use the sign function with scaling vectors [6, 7, 27]. The accuracy gap for the ImageNet benchmark is against real-value networks is around a 16% of top-1 accuracy when using this strategy; the effect on binarization for face recognition purposes has not been explored. Producing robust face descriptors using binary intermediate representations remains as an opportunity area.

Current approaches for Lightweight Face Recognition have been mostly based on regular Lightweight CNNs, with no approaches based on Binary and Quantized Networks for HR and VLR face recognition scenarios. In this paper, we introduce BinaryFaceNet, the first approach for Binarized VLR Face Recognition, inspired in Lightweight and BNN designs. We make the following contributions:

- Propose a new BNN architecture for very low resolution face recognition, BinaryFaceNet, achieving an efficient and robust face recognition in VLR face recognition scenarios.
- Improved efficiency over state-of-the-art Lightweight and Binary CNNs, specifically tailored for face recognition.
- A thorough analysis and discussion of the design choices necessary for implementing extremely lightweight Binarized CNNs for VLRFR applications.

This paper is organized as follows: Section 2 explains and discusses the related works to efficient face recognition models and binarization. Section 3 details our BinaryFaceNet approach to efficient very low resolution face recognition. In Section 4, we present the results of BinaryFaceNet on state-of-the-art VLR benchmarks, while Section 5 discusses our findings and the limitations of our method. Section 6 provides the final remarks of our work.

2. Related Work

In this section we will describe and discuss the methods in the state of the art for Lightweight CNNs and BNNs for face recognition. In the particular context of VLR face recognition, a study conducted by Martínez-Díaz et al. [24] provided insights for training efficient lightweight CNNs on VLR benchmark datasets, showing good results using inter-area interpolation as single pre-processing for fine-tuning the networks, with better results when combining interpolation methods. Lighweight CNNs generally rely on more efficient operations such as 1×1 point-wise convolutions to reduce channel depth, inverted bottleneck structures to reduce parameters, and favoring low-dimensional embedding outputs. BNNs favor lower-bit representations while retaining information using residual connections and adding learneable parameters to the quantization step.

2.1. Lightweight CNNs for Face Recognition

As previously mentioned, the efficient Lightweight approaches for face recognition are based on efficient methods for general-purpose computer vision tasks. MobileFaceNet is based on MobileNetV2 [28]. The authors used PReLU activations instead of ReLU, replaced the Global Average Pooling layer with Global DepthWise Convolution, and made use of a 1280-D embedding representation. ShuffleFaceNet [23] is based on the ShuffleNetv2 [20] architecture. The authors propose to enhance face recognition performance by also replacing ReLU with PReLU, use a fast downsampling strategy with a 2-strided convolution after receiving the input data, and replacing the Global Average Pooling layer with a Global DepthWise Convolution. They also use a 128-Dimensional feature vector as output, showing that it is possible to have an efficient and accurate face descriptor with fewer dimensional units than in the state-of-the-art solutions.
VarGFaceNet [32] takes the VarGNet as a baseline model [36], making changes to improve face recognition performance. In their normal block, they add a Squeeze-and-Excitation operation before adding the regular flow signal with the input signal, and added a PReLU activation before the output. In the embedding setting, VarGFaceNet uses 1×1 convolutions to expand the dimension of the feature map to a 1024 channels. Afterwards, they reduce the spatial dimensions to 1×1 with grouped and pointwise convolutions. Their output is a 512-Dimensional descriptor.
MixFaceNets is based on MixNets [30]. The authors choose to firstly downsample the representation with a 2-strided convolution and add a residual block in the head block, then added a MixConv block with different kernel sizes (3, 5, and 7) to downsample the representation. Finally, in the embedding setting, they expand the representation depth to 1024 channels, and output a 512-Dimensional descriptor using a Global DepthWise Convolution.

2.2. Binary Neural Networks

Scalable BNN approaches such as ABC-Net [17], DoReFa-Net [37], and BinaryDenseNet [2] stack binary convolutional layers attempting to approximate real-valued layers, in a residual-layer fashion. This often incurs in efficiency penalties as the number of operations gets closer to

the real-value version. Balancing the architecture hyperparameters and layer design is key to binarized approaches in order to preserve the best cost-to-performance ratio. Furthermore, to the best of our knowledge, most of the binary network approaches propose to estimate one real-value CNN in particular (ResNet [12], most commonly). In Quicknet [1], the authors propose to use Quantized Depthwise and Quantized Separable Depthwise convolutions with 3×3 kernel size to alleviate the full convolutional operation overhead and maintaining an expressive channel filtering. They use a reduced 16-Dimensional depth channel intermediate representation and full-precision layers for spatial downsampling. They also avoid a heavy use of residual connections to avoid potential latency.
In VLR face recognition literature, Super Resolution (SR) is a prominent approach. Some SR methods using BNNs have been explored, such as Super Resolution Binarization (SRB) [21] and Binarized Single Image Super Resolution (BSISR) [31]. In SRB, the authors proposed and reported good results when adding a learnable α scaling factor for binarized layers. In BSISR, the authors proposed a bit accumulation mechanism, which sums scaling factors α for binarized layers and β for binarized activation layers, for three consecutive layers. An attempt to approximate floating-point residual convolutional blocks using grouped binary residual blocks was proposed in Group-Net [38].

2.3. Remarks over the state of the art

Analyzing the state of the art, we note that the main pass for binarizing weights is to use the Straight-Through-Estimator sign function. This does not take into account the median of the distribution of the data which might be a determining factor to better approximate real values into binary values. This is partially mitigated by [17] by using different activation scales. The scaling factors α and β are linear by default. Adding non-linearity is crucial to get better performance, as showed in [37].

Furthermore, Binary convolution blocks are often put in place for standard convolutions, which improves efficiency. An attempt to approximate one floating-point residual convolutional blocks using binary residual blocks was proposed in [38]. However, there is a gap for specialized binary architecture building blocks akin to lightweight convolutional neural networks. Sharing information between channels, which is what makes a strong case for homogeneous feature extraction for very low resolution face recognition, is not present in binary convolutional units, other than in [1].

Many BNNs use a binarization step consisting in Batchnorm → Activation → Sign function → Pooling; where pooling layers have proven to be ineffective for CNNs as evidenced in Lightweight CNNs, favoring Strided convolutions for downsampling. This avoids further loss of information by having the information present in the downsampled representation instead of completely eliminating data in the pooling step. Also, locality has not been fully exploited in current binarization block designs, other than in [31]. This local information sharing is paramount for applications in face recognition and subspace projection mapping.

3. Approach

In this section, we present in detail our approach named BinaryFaceNet. BinaryFaceNet is based on a block architecture, scalable in depth with a depth channel hyperparameter in the network definition. As per our analysis from the previous section, we found a gap for specialized binary architecture building blocks akin to Lightweight CNNs for face recognition. The main inspiration for our approach are the design guidelines from MobileFaceNet [3], VarGFaceNet [32], and Group-Net [38]. We emphasize sharing information between channels, which is what makes a strong case for homogeneous feature extraction for face recognition.

3.1. General architecture

The network is organized in a Head block, a Global Feature Extraction (GFE) block, two Local Feature Extraction (LFE) blocks, and an Embedding block. We established the following design guidelines for our method:

- We avoid pooling layers in favor of using greater strided convolutions when we need to decrease the feature map size.
- Using a Squeeze-and-Excitation (SE) module in the head block, calculating the SE factor using progressive downsizing via quantized convolutions.
- Using BatchNorm and PReLU with floating-point precision as an activation function after convolutional layers.
- Adding alternating residual connections (Adds) in the block designs to compensate for the heavy signal degradation from the quantized layers.
- Grouped convolutions with 2-step strides for efficiently downsampling the feature representations.
- Using the DoReFa quantizer at every quantized layer for adding non-linearity to the quantization and optimization steps, further stability, and k-bit flexibility to the representation.
- Using a compact 128-Dimensional vector as feature output.
- Using Quantized DepthWise(QDepthWiseConv) and Quantized Separable (QSeparableConv) convolutions to share channel-wise filters.

Our basic Quantized Convolutional block (QConvBlock) is defined with a Binarized Convolution followed by a BatchNorm layer and a PReLU activation. The BatchNorm layer aids in stability and convergence of the network. For

the network dimensions, we take an approach similar to MobileFaceNet with 2-strided convolutions. Table 1 describes the overview of our downsampling methodology in our method.

Name	Input	Output
Input layer	3×112×112	3×112×112
Head Block	3×112×112	64×56×56
GFE Block	64×56×56	64×56×56
LFE Block 1	64×56×56	64×28×28
LFE Block 2	64×28×28	64×14×14
Embedding Block	64×14×14	128×1×1

Table 1. General overview of the steps and feature map sizes in BinaryFaceNet. We first expand to 64 channels in the head block and downsize the feature map, then process a high-resolution 56x56 feature map in the General Feature Extraction (GFE) block, and further dowsize the representation at the start of the Local Feature Extraction (LFE) blocks. We finalize with the Embedding block downscaling and expanding the representation channels to a 128-Dimensional vector using strided convolutions.

3.1.1 Head block

The main challenge of BNN is signal degradation. To improve the discriminative ability further down the network, we maintain the input floating-point representation for retaining the details in the input signal as much as possible. Figure 1 details the operations in the head block. Our Head block expands the representation to 64 channels, then calculates the SE weight factor for a weighted floating-point feature map subsequently. However, we avoid using pooling operations due to their reduced discrimination ability and use a Reduction block to favor quantized grouped strided convolutions to retain more information and extract more accurate scaling factors to the data while downsizing. Finally, we scale the data with the output from the SE block and downscale the representation using a 2-strided convolution to be passed down along the network. We choose to use grouped convolutions heavily in this step due to their discrimination ability while increasing the efficiency when downsizing the representation.

3.1.2 Global Feature Extraction block

Our Global Feature Extraction block, shown in Figure 2, works with the same representation size throughout. This allows us to combine the input signal with the residual of two posterior convolutional operations. This block processes HR feature maps with a resolution above 32×32. Similar to QuickNet [1], we use Quantized DepthWise Convolutions (QDepthWiseConv) for efficient channel filtering through the depth dimension of the representation. We choose the QDepthWiseConv operation to reserve more

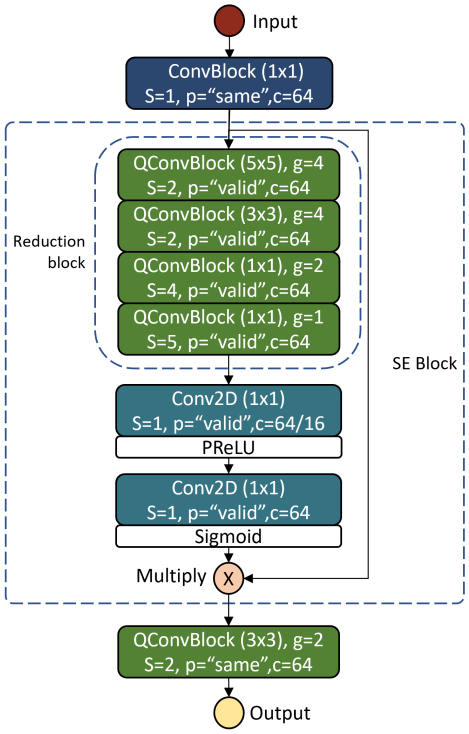

Figure 1. Head block design of BinaryFaceNet. We propose to use a Reduction block with QConv operations inside the Squeeze-and-Excitation (SE) block for calculating the weights, instead of a pooling layer. We use strided (2,2,4,5 sequentially) convolutions with different kernel sizes (5,3,1,1 sequentially) and grouped convolutions to alleviate the operations overhead.

complex operations for the other building blocks in our architecture while still sharing filters through the channel dimension. We use a PReLU layer to re-scale the sum of the feature maps. The input signal is introduced back after two convolutional operations and the signal from the first convolution gets reintroduced after a third single convolutional layer. This helps to compensate the signal degradation from the quantization process at different processing points. The compensated signal before activation is later reintroduced after a QDepthWiseConv layer and re-scaled. This last signal is the output of the GFE block for posterior processing at the LFE blocks.

3.1.3 Local Feature Extraction block

The Local Feature Extraction block, illustrated in Figure 3, firstly downsizes the representation with a non-grouped 2-strided convolution. We choose to use Quantized Separable Depthwise Convolution layers (QSeparableConv) instead of the simpler QDepthWiseConv layers from the previous GFE block. The QSeparableConv performs additional point-wise convolutions separating spatially the depth

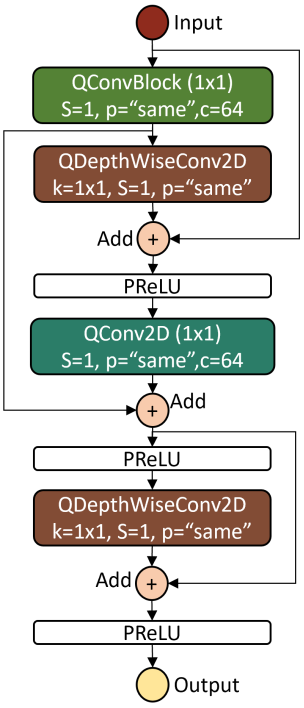

Figure 2. Global Feature Extraction block in our architecture, for higher resolution feature maps. We use residual connections from the signals from the input, the first QConvBlock, and after the second addition operation. This mitigates the heavy signal degradation from quantization. We choose to use QDepthWiseConv operations for depth-wise filtering to add robustness and without the overhead of pointwise convolutions.

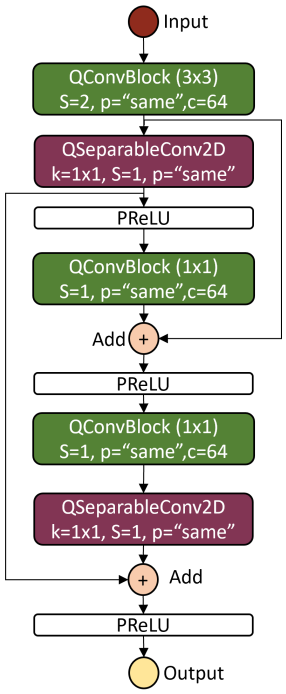

Figure 3. Local Feature Extraction block design of BinaryFaceNet. This module processes VLR feature maps and features more operations than the GFE block. It reduces the representation size by half with a 2-strided convolution at the start of the block and propagates that signal to add it after the second QConvBlock. The signal from the two QSeparableConv layers are added at the end of the block.

kernel after a QDepthWiseConv, but is less expensive than a regular convolutional layer. After a PReLU activation layer, we use another QConvBlock and we reintroduce the signal from the initial downsampling back to the feature map flow. After re-scaling with PReLU, we further process the feature map by using another QConvBlock followed by a QSeparableConv. Finally, we add the signal from the first QSeparableConv layer back to the network and re-scale for the final block output.

3.1.4 Embedding block

In our embedding block, detailed in Figure 4, we define our face descriptor as a 128-Dimensional vector. A small face descriptor, such as this one, is efficient and adequate, shown to perform adequately in face recognition performance benchmarks [23, 24]. We downscale the representation using two serialized full-precision convolutional blocks, favoring strided convolutions and further avoiding pooling operations, with a kernel size of (3,3) and (4,4), and strides of 3 steps and 1 step, respectively. When using a low-dimension output vector, we are optimizing the computations in the posterior Fully Connected (FC) layers or any other posterior verification and matching calculations.

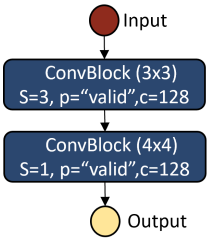

Figure 4. The Embedding block consists of two straightforward full-precision Convolution blocks with "valid" padding, filtering the full area of the feature map with no additional padding. The channel dimension is expanded to 128 at the start of the block and is consistent throughout.

3.2. Training settings

In this work, we performed experiments of our proposed BinaryFaceNet against other state-of-the-art BNNs: QuickNet [1] and BinaryDenseNet [2], with different complexity configurations for BinaryDenseNet (depths 28, 37, and 45).

For training, we used the LARQ framework on Keras over Tensorflow. We use the SGDW optimizer with batch sizes of 128 as per our GPU memory limitations for any given model. For our hardware settings, we use one NVIDIA GeForce GTX 1080Ti 11GB VRAM GPU for training. The learning rate is set to 0.01 for BinaryFaceNet for stability; the rest of the networks use a 0.1 value. The learning rate is adaptative, multiplied by 0.1 at iterations 9, 15 and 18. The optimizer parameters are set to 0.9 for the momentum and 5e-4 for the weight decay. We use the ArcFace loss function [8], which is highly optimized for face recognition performance, selecting a scaling value of 16 for all BNNs, tested empirically for stability. We use an input resolution of 112×112×3 and 128-Dimensional embeddings for all BNNs.

3.2.1 Pre-processing and datasets

As per studies in the VLR FR area [24], mentioned in previous sections, we mostly employ inter-area interpolation to simulate the conditions of VLR images as a single method for interpolation, then up-scaling to the input-layer size of 112×112.

We perform pre-training on the MS1-Celeb-1M dataset [11] and fine-tuning on the CASIA WebFace [33], SCface [10], QMUL-TinyFace [4], and QMUL-Survface [5] datasets. All of these datasets, except the CASIA WebFace, contain VLR images of 32×32 pixels or less natively. We upscale all the input images to 112×112 using inter-area interpolation for all datasets except the CASIA WebFace, for which we use bicubic interpolation as well. The input images are normalized to a [-1, 1] range by subtracting the mean pixel value of 127.5 and dividing by 128.

The SCface dataset contains 130 subjects and 4,160 images from different surveillance camera distances: d1(4.2m), d2(2.6m) and d3(1.0m) with native VLR images and their corresponding subject's HR gallery counterpart.

The CASIA WebFace dataset is an unconstrained face recognition dataset synthetically downscaled to LR images of 7×7, 14×14, 28×28 and 56×56 using bicubic and inter-area subsampling strategies. It contains 10,575 identities and 494,414 images in total.

The QMUL-Survface contains native surveillance images from 15,573 subjects and 463,507 images. The input resolutions vary but are always below 32×32 pixels. The QMUL-TinyFace dataset contains unconstrained face images from the web with 5,139 identities and 169,403 images.

For fine-tunning on the Tinyface and Survface datasets, we combined them into a single one by matching the identities. We test the two individual datasets with their own test set under this trained model on the combined dataset.

The MS-Celeb-1M dataset contains a web compilation of over 10 million HR face images from over 100K identities gathered from the web. We used the cleaned version of this dataset (also known as MS1M-V3 or MS1M-RetinaFace), the baseline of the LFR@ICCV19 lightweight face recognition challenge [9].

4. Results

In this section, firstly we present the computational complexity of our face recognition model against state-of-the-art BNNs for general-purpose computer vision tasks: Quicknet [1] and BinaryDenseNet [2] with depth configurations 28, 37, and 45. Secondly, we show our face recognition results on traditional face recognition benchmarks and VLR-specific datasets using those same BNNs, except for BinaryDenseNet45.

4.1. Computational complexity and inference time performance

In Table 2 we report the number of binary Multiply-Accumulate (MAC) operations for 1-bit and Full Precision 32-bit representations, the model size in Megabytes(MB), the total number of parameters and the runtime on the Larq Compute Engine (LCE) [1] benchmark tool in single-threaded performance, running on an NVIDIA Jetson Nano 2Gb with an ARM A57 processor. The inference time benchmark from LCE represents the time it takes for each method to perform a forward pass under the aforementioned setting.

With our BinaryFaceNet architecture, we perform 57% less binary MACs and 44% less floating-point 32-bit MACs, compared against QuickNet. The 1-bit MAC operations gap is wider against BinaryDenseNet designs: of 79%, 83%, and 88% on network depths 28, 37, and 45, respectively. In terms of model size, our method has a memory requirement of only 3.8MegaBytes, with the rest needing 44% or more space to run. Our design also runs with only 506K parameters, representing only 3.9% of the total parameters for QuickNet (12.7M) and 11% of the total parameters of BinaryDenseNet 28 (4.5M). Most importantly, for inference time per image in a practical setting using LCE, BinaryFaceNet shows a 91% runtime speedup against QuickNet [1].

The presented efficiency results point to the feasibility of implementing a real-time face recognition application on an affordable embedded device like the NVIDIA Jetson Nano. Our method is the only one capable of achieving 6 to 7 Frames Per Second (FPS) using only one thread of the ARM A57 CPU. In this scenario, we can off-load the rest of the application and FR pipeline processing burden to the rest of the ARM A57 CPU cores and the included GPU.

Method	1-bit MACs	32-bit MACs	Mem. (MB)	Total Params	Time img/s
BinaryDenseNet45 [2]	1.59G	59.7M	4.2	13.1M	6.2s
BinaryDenseNet37 [2]	1.12G	48.9M	2.6	8.0M	4.4s
BinaryDenseNet28 [2]	893.2M	49.99M	1.8	4.5M	3.7s
QuickNet [1]	431.7M	6.8M	2.21	12.7M	1.8s
BinaryFaceNet	**184.9M**	**3.8M**	**1.3**	**506K**	**0.16s**

Table 2. Efficiency for the state-of-the-art Binary Neural Networks, for a 112×112 input and 128-Dimensional feature output. Our ultra-compact model BinaryFaceNet is the most efficient in design and inference runtime in the Larq Compute Engine [1] single-threaded benchmark on the NVIDIA Jetson Nano ARM A57 processor.

4.2. VLR Identification and Verification Performance

In this subsection we present our experiments with the training settings described on Section 3.2.

4.2.1 Experiments on the SCface dataset

We present the identification results for the SCface dataset [10] according to the protocol used in the state of the art [24, 34, 35], using 50 randomly chosen subject identities for training and 80 for testing. Table 3 shows the recognition rates for the three distances of the subjects to the camera, as defined in the dataset (4.2m, 2.6m, and 1.00m).

For this identification test, BinaryFaceNet performs with a higher accuracy performance on the hardest setting (4.2m) where the camera is the farthest from the subject, compared to the next most efficient state-of-the-art BNNs. The method has higher performance drops on the 2.6m and 1.0 settings, where we start seeing the limitations of our approach. This shows the effectiveness of our block design for processing the very low resolution feature maps and the compromise on capturing higher density details when balancing the accuracy-efficiency trade-off.

4.2.2 Experiments on the TinyFace and SurvFace datasets

For the combined TinyFace and SurvFace dataset, we present the identification results for TinyFace in Table 4 and the verification results for SurvFace on Table 5.

In the TinyFace identification setting (Table 4), our method scores an accuracy drop across the Rank-1, Rank-20, and Rank-50 metrics of 18% in the hardest setting and 20% in the other two settings respectively, against QuickNet. This gap is reduced against both BinaryDenseNet approaches in all settings, which are considerably more expensive methods.

In the SurvFace verification results (Table 5), our method achieves a similar drop against Quicknet, showing a drop of 17.3% on the TAR@FAR=1% setting and 9.8% on the TAR@FAR=0.1% setting, with this latter one being the harder verification setting. In a Mean Accuracy metric, our method achieves only a 5% drop against the next most efficient approach as well.

4.2.3 HR-VLR verification using inter-area interpolation on LFW

Running our benchmarks for simulated VLR verification scenarios, as also presented in in [24], we present Table 6, showing the verification rates for our simulated HR matching with VLR probe scenario with the CASIA WebFace and SCface datasets, respectively. The VLR probes are images from the LFW dataset downscaled using inter-area interpolation. This experiment aims to test cross-dataset verification performance with the simulated VLR conditions provided by the interpolation methods.

Our proposal BinaryFaceNet achieves a 7.5% accuracy compromise on the hardest case against QuickNet, when fine-tunning in the CASIA WebFace and a superior verification performance in the same case when fine-tunning using the SCface dataset. This shows a compelling result for the descriptor generated by BinaryFaceNet, especially for verification performance on affordable hardware.

5. Discussion

In this section, we discuss the outcomes of our research work in terms of training ultra-compact binarized models, limitations on hardware implementations, and balancing efficiency and accuracy; while providing insights for future research directions. We primarily discuss these topics from a real-time application perspective using affordable hardware.

5.1. Limitations on hardware implementations

Computer Vision technology in real-world applications is usually dependant on the target hardware constraints. Our target hardware for this proposal is the NVIDIA Jetson Nano, an affordable inexpensive device in terms of power efficiency. In theory, BNNs should be extremely efficient [27], however, we are currently constrained by the

Method	d1 (4.2m)	d2(2.6m)	d3 (1.0m)
BinaryDenseNet-37 [2]	38.0%	79.2%	86.7%
BinaryDenseNet-28 [2]	35.0%	79.2%	89.0%
QuickNet [1]	43.2%	**83.0%**	**91.2%**
BinaryFaceNet (ours)	**48.7%**	70.5%	60.5%

Table 3. SCface identification identification performance using the 50-80 subject testing protocol [24]. Our method outperforms the rest of the BNN methods on the hardest cases (d1 at 4.2m).

Method	Rank-1	Rank-20	Rank-50	mAP
BinaryDenseNet-37 [2]	39.1%	56.2%	61.6%	31.3%
BinaryDenseNet-28 [2]	37.5%	56.2%	61.3%	30.0%
QuickNet [1]	**40.9%**	**59.0%**	**64.5%**	**32.7%**
BinaryFaceNet (ours)	22.9%	38.0%	43.5%	16.9%

Table 4. QMUL-TinyFace identification experiments using our combined TinyFace and SurvFace dataset. Our method achieves a consistent performance compromise on real-world datasets against other BNNs, while taking less than 10% of the inference time required for those methods.

software frameworks supported for our hardware configuration. In this project we used the LARQ Compute Engine [1] for its support for state-of-the-art BNN operations such as grouped binary convolutions, DoReFa quantizers, and its highly-efficient custom binary ops. On the other hand, this platform only supports ARM and xCORE architectures. Even though mainstream x86 architectures are affordable and available, binarized convolutional operations are not optimized for this platforms. This also means that embedded platforms with onboard GPU modules cannot be utilized due to the binary framework only working on ARM CPUs. Additional research efforts are needed to broaden the availability of robust BNN technology on more mainstream platforms.

In order to close the gap for a complete VLR FR application to run in such power-efficient devices, we must consider the computational effort of running the full face recognition pipeline: camera image processing, face detection, alignment, feature extraction, and matching. As previously mentioned, BinaryFaceNet is the only one capable of running in a VLR FR application close to a real-time setting (6 to 7 FPS, without overclocking) in a single ARM core with comparable accuracy performance to the state-of-the-art BNNs while reserving the rest of the CPU cores and GPU for the rest of the operations in the FR pipeline. We attribute the inference time speedup of our method to the combination of drastically reducing the number of network parameters and total MAC operation count. Reducing the total parameter count by at least 89%, maintaining most of our network binarized (over 97% of total MAC operations), and reducing the 1-bit MAC count by 57% and the 32-bit MAC count by 44% against other BNNs has allowed us to achieve inference performance closer to real-time scenarios.

5.2. Trade-off between complexity and accuracy

The BinaryFaceNet proposal in this paper is particularly optimized to our hardware constraints and VLRFR scenario. The face descriptor generated by BinaryFaceNet is a step towards a more efficient representation for a cross-dataset verification scenario. The BinaryFaceNet architecture can also benefit from more hardware resources available by tuning network hyperparameters; particularly the internal depth size and the quantization bits. The quantization bits from the reduction block inside the head block (Figure 1) is one of the primary points for balancing for efficiency and accuracy performance. Our quantizer choice, DoReFa [37], allows for an easy k-bit adjustment. Balancing this quantized k-bit representation can lead to a more robust representation since it directly affects the calculation of the SE weight that scales the input image after the first full-precision convolution. This opens the door for efficient face recognition research using this BNN design in other scenarios such as heterogeneous FR (sketches, NIR-VIS, etc.), aging subjects, and racial bias.

5.3. Training ultra-compact face recognition binarized stable models

In our experiments, we found several hyperparameters and training settings to appropriately work on the face recognition scenario. The recommended state-of-the-art optimizer for training BNNs using general-purpose datasets is the Adam optimizer [22], however, in our experiments with face recognition datasets we found this strategy to be ineffective when optimizing the state-of-the-art FR loss function ArcFace [8]. As such, we employed the SGDW and achieved the reported performance in the previous section. Furthermore, the ArcFace loss scale hyperparamter s was

Method	TAR@FAR		AUC	Mean
	1%	0.1%		Acc.
BinaryDenseNet-37 [2]	**49.1%**	21.0%	**91.7%**	**84.2%**
BinaryDenseNet-28 [2]	44.7%	**22.1%**	91.3%	83.7%
QuickNet [1]	47.3%	18.1%	91.6%	83.9%
BinaryFaceNet	30.0%	8.3%	87.1%	78.6%

Table 5. QMUL Survface verification performance when fine-tunning with the combined TinyFace and Survface dataset. In this scenario, our method has less of a performance compromise comparatively, with only a 9.8% drop for TAR@FAR=0.1% and competitive results on AUC and overall mean accuracy on this real-world surveillance benchmark.

Method	CASIA FT - LFW			SCface FT - LFW	
	7×7	14×14	28×28	7×7	14×14
BinaryDenseNet-37 [2]	**90.2%**	95.3%	**97.7%**	56.6%	**79.3%**
BinaryDenseNet-28 [2]	**90.2%**	95.0%	97.5%	55.3%	78.1%
QuickNet [1]	89.1%	**94.5%**	97.1%	53.9%	75.2%
BinaryFaceNet (ours)	81.6%	89.2%	92.8%	**58.7%**	78.1%

Table 6. Verification rates for the simulated HR matching with VLR probes from the LFW dataset using inter-area interpolation, when fine-tuning using the CASIA WebFace and SCface datasets separately. The BinaryFaceNet descriptor achieves a very competitive cross-dataset verification performance, outpeforming other BNNs with fine-tunning using SCface, a dataset with native VLR iamgery.

adjusted to 16 from its usual empirical value set to 64. The scale hyperparameter affects the vector scaling in the projected subspace, as such, a larger value leads to higher gradients. This, in turn, also leads to more instability when training BNNs. In the same fashion, the training stability is sensitive with the learning rate hyperparameter. With reduced network designs, such as the case in BinaryFaceNet, the learning rate must be reduced to avoid exploding gradients. The quantizer choice is also important in the design of ultra-compact networks, since it influences training stability and accuracy performance. The DoReFa quantizer leverages non-linear functions (*tanh*) for scaling and gradient clipping, making it an appropriate choice for our ultra-compact model. Further research in hyperparameter tuning for FR scenarios using ultra-compact BNNs is encouraged to further maximize the efficiency benefits of BNN technology.

6. Conclusion

In this paper, we proposed BinaryFaceNet, the first BNN architecture proposal for real-time very low resolution face recognition. We achieved limited accuracy compromises against other state-of-the-art BNNs while performing inference at 6-7 FPS on a single-core ARM CPU setting. Furthermore, we provided a discussion on the limitations of BNN implementations, the trade-off between complexity and accuracy, and the particularities of training stable ultra-compact BNNs for face recognition.

References

[1] T. Bannink, A. Bakhtiari, A. Hillier, L. Geiger, T. de Bruin, L. Overweel, J. Neeven, and K. Helwegen. Larq compute engine: Design, benchmark, and deploy state-of-the-art binarized neural networks, 2020. 3, 4, 5, 6, 7, 8, 9

[2] J. Bethge, H. Yang, M. Bornstein, and C. Meinel. Binarydensenet: Developing an architecture for binary neural networks. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1951–1960, 2019. 1, 2, 5, 6, 7, 8, 9

[3] S. Chen, Y. Liu, X. Gao, and Z. Han. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. *Lecture Notes in Computer Science*, pages 428–438, 2018. 1, 3

[4] Z. Cheng, X. Zhu, and S. Gong. Low-Resolution Face Recognition. *arXiv preprint arXiv:1811.08965*, pages 1–16, nov 2018. 6

[5] Z. Cheng, X. Zhu, and S. Gong. Surveillance face recognition challenge. *arXiv preprint arXiv:1804.09691*, 2018. 6

[6] M. Courbariaux and Y. Bengio. Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1. *CoRR*, abs/1602.02830, 2016. 1, 2

[7] M. Courbariaux, Y. Bengio, and J. David. Binaryconnect: Training deep neural networks with binary weights during propagations. *CoRR*, abs/1511.00363, 2015. 1, 2

[8] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4685–4694, 2019. 6, 8

[9] J. Deng, J. Guo, D. Zhang, Y. Deng, X. Lu, and S. Shi. Lightweight face recognition challenge. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 1, 6

[10] M. Grgic, K. Delac, and S. Grgic. Scface - surveillance cameras face database. *Multimedia Tools Appl.*, 51:863–879, 02 2011. 6, 7

[11] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, page 87–102, Cham, 2016. Springer International Publishing. 6

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

[13] K. Helwegen, J. Widdicombe, L. Geiger, Z. Liu, K.-T. Cheng, and R. Nusselder. Latent weights do not exist: Rethinking binarized neural network optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 1

[14] 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 *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1314–1324, 2019. 1

[15] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. 1

[16] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡1mb model size. *ArXiv*, abs/1602.07360, 2017. 1

[17] X. Lin, C. Zhao, and W. Pan. Towards accurate binary convolutional neural network. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 344–352, Red Hook, NY, USA, 2017. Curran Associates Inc. 2, 3

[18] Z. Lu, X. Jiang, and A. Kot. Deep Coupled ResNet for Low-Resolution Face Recognition. *IEEE Signal Processing Letters*, 25(4):526–530, 2018. 1

[19] L. S. Luevano, L. Chang, H. Méndez-Vázquez, Y. Martínez-Díaz, and M. González-Mendoza. A study on the performance of unconstrained very low resolution face recognition: Analyzing current trends and new research directions. *IEEE Access*, 9:75470–75493, 2021. 1

[20] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1, 2

[21] Y. Ma, H. Xiong, Z. Hu, and L. Ma. Efficient super resolution using binarized neural network. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 694–703, 2019. 3

[22] B. Martinez, J. Yang, A. Bulat, and G. Tzimiropoulos. Training binary neural networks with real-to-binary convolutions, 2020. 8

[23] Y. Martinez-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 *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 1, 2, 5

[24] Y. Martínez-Díaz, H. Méndez-Vázquez, L. S. Luevano, L. Chang, and M. Gonzalez-Mendoza. Lightweight low-resolution face recognition for surveillance applications. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5421–5428, 2021. 1, 2, 5, 6, 7, 8

[25] J. Memon, M. Sami, R. A. Khan, and M. Uddin. Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). *IEEE Access*, 8:142642–142668, 2020. 1

[26] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3523–3542, 2022. 1

[27] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. *CoRR*, abs/1603.05279, 2016. 1, 2, 7

[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2

[29] C. D. Suarez-Ramirez, M. Gonzalez-Mendoza, L. Chang, G. Ochoa-Ruiz, and M. A. Duran-Vega. A bop and beyond: A second order optimizer for binarized neural networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1273–1281, 2021. 1

[30] M. Tan and Q. V. Le. Mixconv: Mixed depthwise convolutional kernels. *CoRR*, abs/1907.09595, 2019. 2

[31] J. Xin, N. Wang, X. Jiang, J. Li, H. Huang, and X. Gao. Binarized neural network for single image super resolution. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, pages 91–107, Cham, 2020. Springer International Publishing. 3

[32] 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 *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 1, 2, 3

[33] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch, 2014. 6

[34] X. Yin, Y. Tai, Y. Huang, and X. Liu. Fan: Feature adaptation network for surveillance face recognition and normalization, 2019. 1, 7

[35] J. Zha and H. Chao. Tcn: Transferable coupled network for cross-resolution face recognition*. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3302–3306, 2019. 1, 7

[36] Q. Zhang, J. Li, M. Yao, L. Song, H. Zhou, Z. Li, W. Meng, X. Zhang, and G. Wang. Vargnet: Variable group convolutional neural network for efficient embedded computing. *ArXiv*, abs/1907.05653, 2019. 2

[37] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients, 2018. 1, 2, 3, 8

[38] B. Zhuang, C. Shen, M. Tan, L. Liu, and I. Reid. Structured binary neural networks for accurate image classification and semantic segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 413–422, 2019. 3