Face Recognition with hybrid binary network

Simone Moro Claudio Marchisio Viviana D'Alto
STMicroelectronics

Introduction

Face Recognition has become a prominent algorithm in everyday life thanks to the remarkable results obtained with the introduction of deep neural networks.
State of the art Face Recognition solutions are based on complex neural networks, which usually require a considerable amount of memory and computational power to be executed; alternatively, the inference could be executed on a cloud service with inevitably privacy concerns.
The possibility to run a Face Recognition solution locally without the need of a powerful GPU and with a reasonable amount of memory would be an added value for many devices.
This work proposes a new network topology to perform Face Recognition on a microcontroller unit, like the STM32H743ZI, equipped with a CPU running at up to 480 MHz, 2 MB of FLASH and 1 MB of RAM.
The main contribution of this work is to use different quantization schemes for different parts of the proposed neural network, in particular the usage of binary quantization is exploited to speed up the inference and reduce the memory requirements for the neural network weights. The deep quantization of the network has been performed using the Qkeras[1] framework, which enables the possibility to customize the bit precision for each layer.

Face Recognition with neural networks

In the context of Face Recognition, neural networks are used to process face images in order to extract the most important features of the input face. These important features of the face are described by an n-dimensional vector, which is called "embedding" and it represents a compact description of the input face; such process is illustrated in Figure 1.

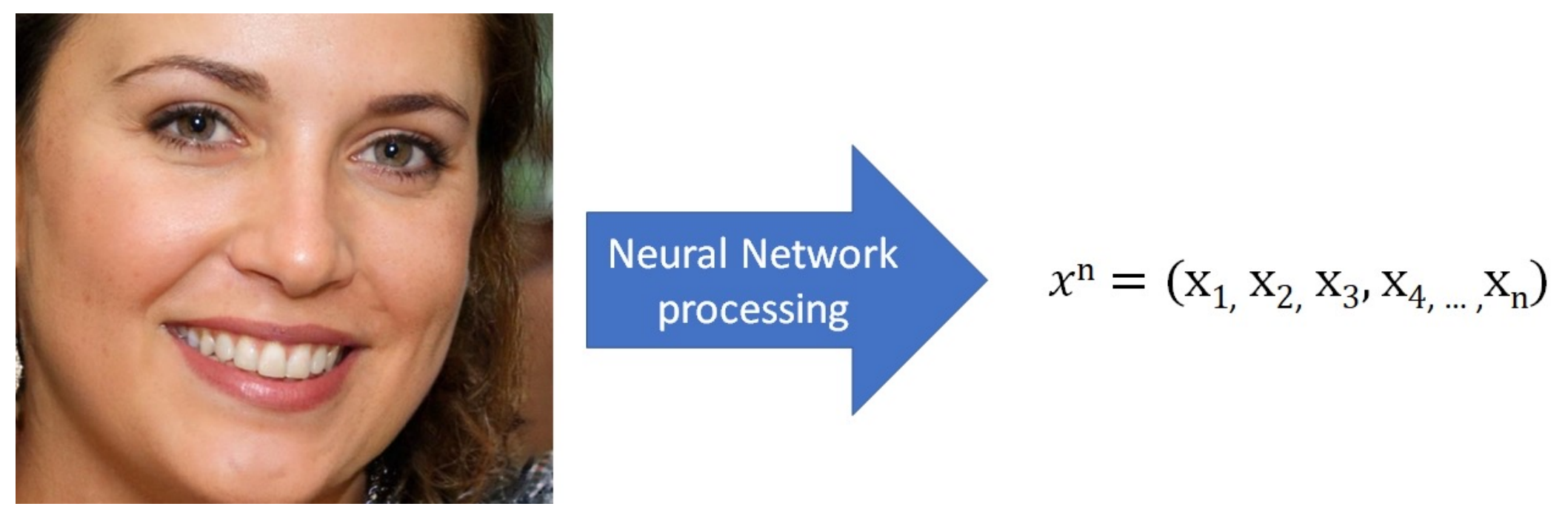
Neural Network processing
$x^n = (x_1, x_2, x_3, x_4, \dots, x_n)$

Figure 1: Generation of the face embedding by a neural network

During the training of the neural network, specific loss functions like Arcface[2] are applied to optimize the embedding generation.

Datasets

The neural networks proposed in this paper have been trained using the MS-Celeb-1M-v1c database, which comprises 3,923,399 images belonging to 86,876 different subjects.
The testing phase of the networks has been performed using the "Labeled faces in the wild" (LFW) dataset, which is made of 13,233 images of 5,749 subjects. The testing procedure implements a pair matching analysis as described on the LFW page.

Image pre-processing

To improve the efficiency of the face recognition neural network, the input faces should be aligned to a frontal pose, where the eyes and the mouth are in a coherent position for all the images.
For this reason, the images are pre-processed to locate the faces inside the frame and to align them to a frontal position.
To perform this job, "Multi-Task Cascaded Convolutional Neural Network" (MTCNN)[3] has been adopted. After alignment, the face is then resized to a fixed resolution of 96x96 pixels. Figure 2 illustrates this process.

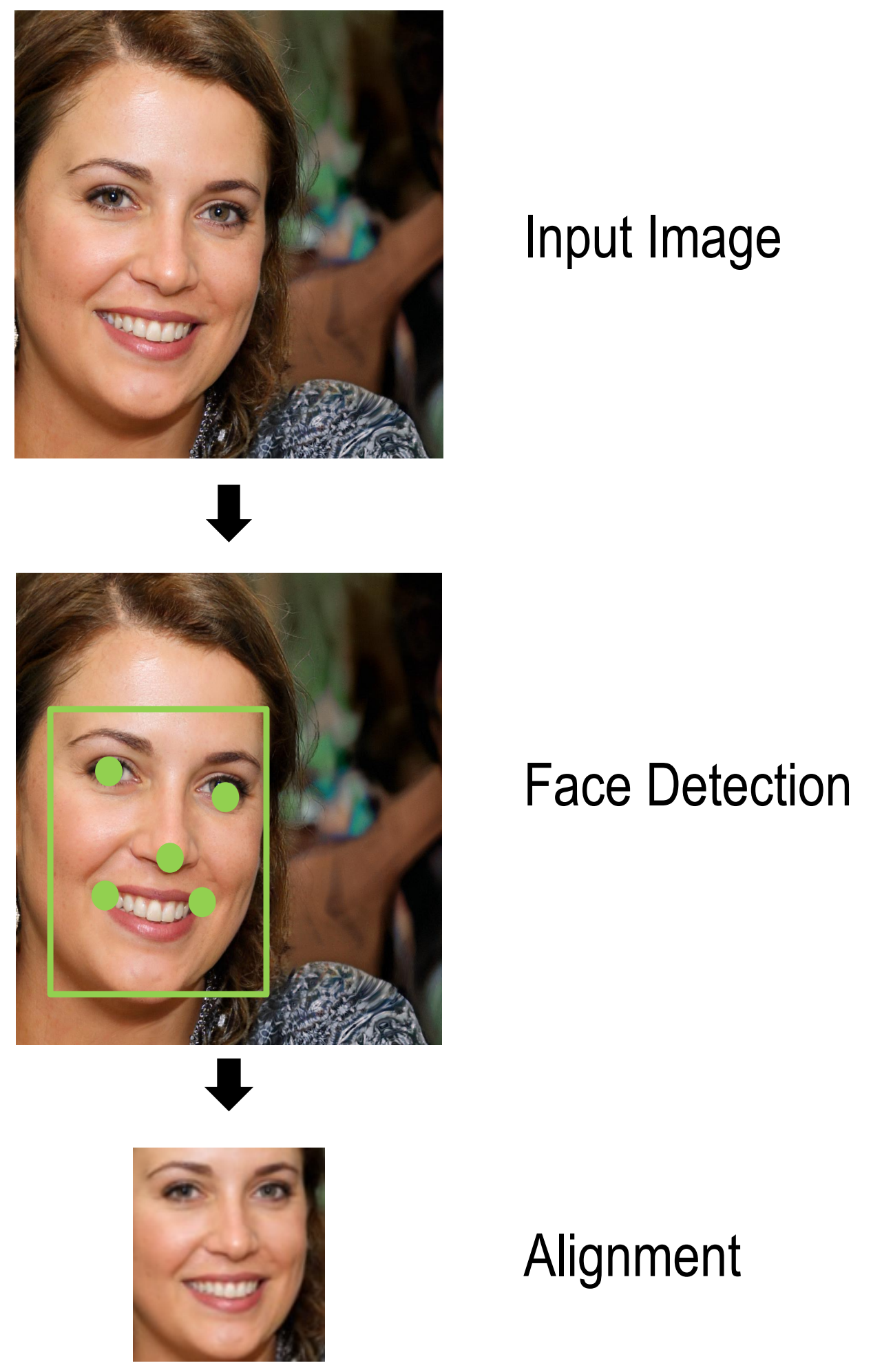

Figure 2: Face Recognition pre-processing

Floating-point and 8-bit baseline

To measure the impact of deep quantization, different topologies have been realized to assess the benchmarks. The first network topology implemented is based on a MobileNetV2 and trained in floating-point precision using the Keras[4] framework. The network has been designed considering a post-training quantization to be applied to match the STM32H743ZI hardware specifications.
The alpha parameter, which controls the deepness or shallowness of the MobileNetV2 has been set to a value of 0.5.
The network has been subsequently quantized with Tensorflow Lite converter and successfully ported to the hardware platform using the STM32Cube.AI plugin of the STM32Cube.MX[5] tool.
This network has been used to benchmark our proposed deeply quantized network.

Network	Flash	RAM	MACC
MNv2_FP	3.27 MB	477 KB	19 M
MNv2_i8	850 KB	163 KB	19 M

Figure 3: Diagram of the main building blocks of the

Binary network topology

A direct binarization of the MobileNetV2 did not produce good results in terms of accuracy, so an investigation has been performed to identify a suitable topology that could retain the accuracy and reduce the memory and the computational cost.
The proposed topology relies on two different blocks of layers, the Residual Block, which performs binary computation and the Transition Block, which performs dimensionality reduction. An illustration of the two blocks can be seen in Figure 4.

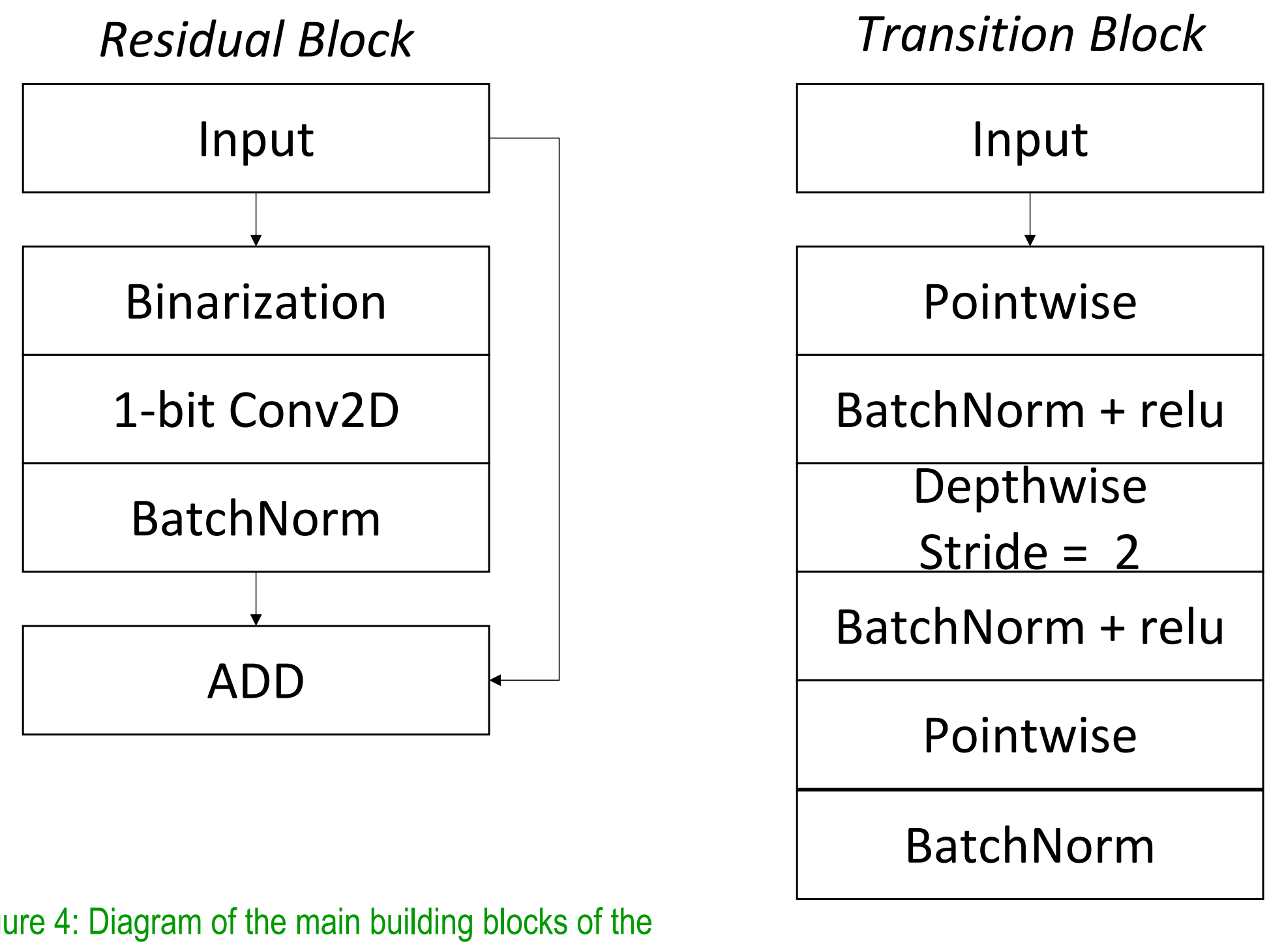

Figure 4: Diagram of the main building blocks of the

The complete topology of the network along with the bit precision per layer is shown in Table 2.

Layer or block	Input resolution	Precision bits
Input	96x96x3	8
Conv2D, s2	96x96x3	8
BN + RELU	48x48x16	8
DepthwiseConv	48x48x16	8
BN + RELU	48x48x16	8
Pointwise	48x48x16	8
BN	48x48x16	8
RESIDUAL BLOCK	48x48x16	1
TRANSITION BLOCK	48x48x24	8
RESIDUAL BLOCK	24x24x24	1
TRANSITION BLOCK	24x24x24	8
RESIDUAL BLOCK	12x12x64	1
TRANSITION BLOCK	12x12x64	8
RESIDUAL BLOCK	6x6x64	1
TRANSITION BLOCK	6x6x64	8
RESIDUAL BLOCK	3x3x128	1
Relu	3x3x128	8
PointWise	3x3x128	8
BN + Relu	3x3x1280	8
DepthWise	3x3x1280	8
BN	1x1x1280	8
Dense	1x1x1280	8
Output	128	32

Table 2: complete network topology

Training and Results

The quantization aware training has been performed with the Qkeras framework using 80% of the dataset for the training and 20% for validation. Figure 5 shows the results of the training for each epoch.

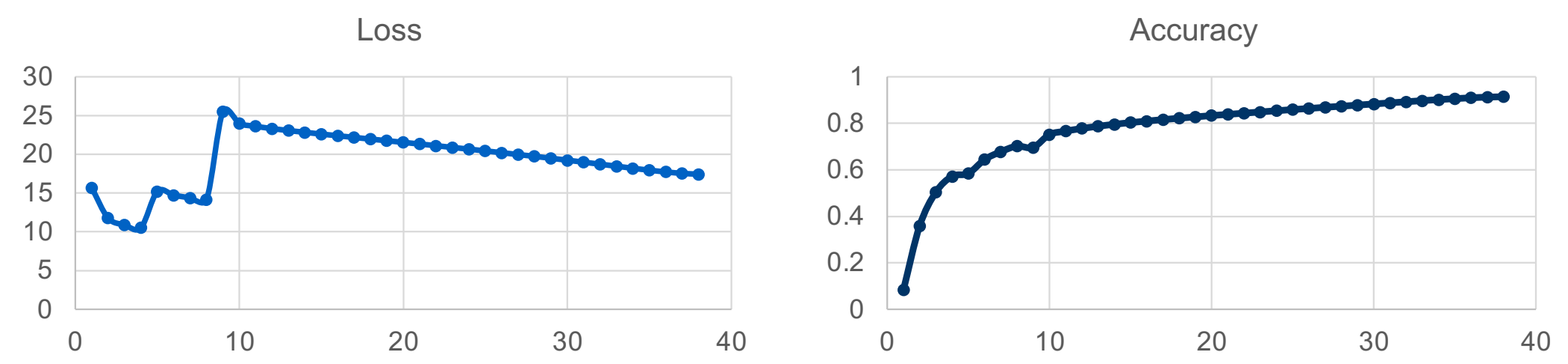

Figure 5: Training loss and accuracy per epoch.

After the training completed, the network has been tested on the LFW dataset. The results are shown in Figure 6, while the memory footprint and the complexity expressed as MACC operations are shown in Table 3.

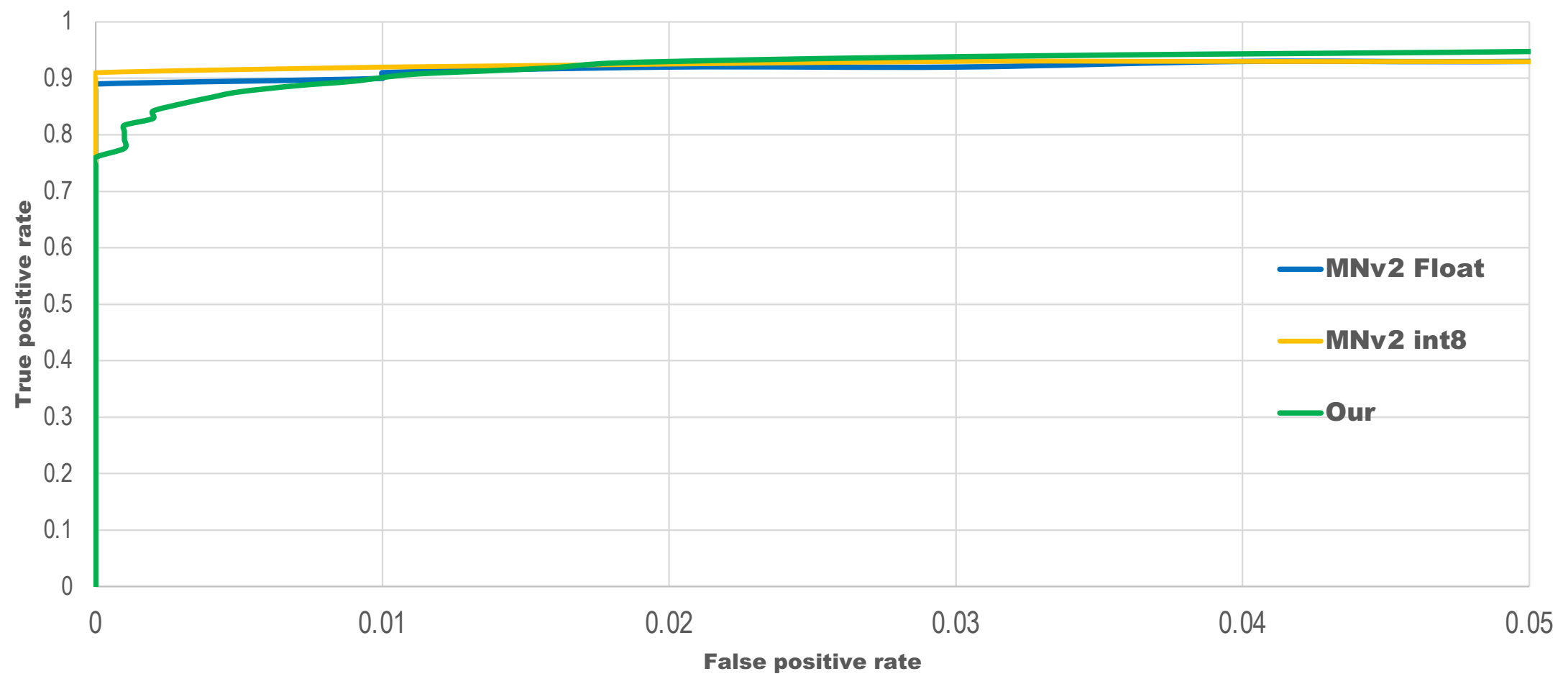

Figure 6: Benchmark of the three topologies using LFW pair matching methodology.

Network	Flash	RAM	MACC	Inference
MNv2_FP	3.27 MB	477 KB	19 M	n/a
MNv2_i8	850 KB	163 KB	19 M	98 ms
Our	**650 KB**	**198 KB**	**26 M**	**89 ms**

Table 3: Memory and complexity results

Is important to note that for the proposed network, despite the higher complexity, the computational time is lower than the int8 network thanks to the efficiency of the binary operations.

Conclusions

Face Recognition is a task that could benefit from a computation at the edge to minimize the power consumption and the privacy concerns. This work shows that binarization of some complex layer of a neural network can lead to improvements in terms of memory consumption and computational power. This optimization enables the possibility to execute the application on low complexity microcontrollers with a minimum loss of accuracy. The future work focuses in further reduce the complexity and the memory footprint trying to keep a high level of accuracy.

References

[1] GitHub -google/qkeras: QKeras: a quantization deep learning library for Tensorflow Keras

[2] Deng J (2018): ArcFace: Additive Angular Margin Loss for Deep Face Recognition

[3] Zhang, K (2016): Joint face detection and alignment using multitask cascaded convolutional networks

[4] Keras: the Python deep learning API

[5] X-CUBE-AI -AI expansion pack for STM32CubeMX - STMicroelectronics