APPLICATION NOTE 7364

FACE IDENTIFICATION USING MAX78000

By: Erman Okman, Gorkem Ulkar

Abstract: The MAX78000 is an ultra-low power Convolutional Neural Network (CNN) inference engine to run Artificial Intelligence (AI) computations on tiny edges of IoT. Yet the device can execute many complex networks to achieve critical and popular applications. This document describes an approach for Face Identification (FaceID) running on the MAX78000 where the model is built with Maxim's development flow on PyTorch, trained with different open datasets and deployed on the MAX78000 evaluation board.

Introduction

Face recognition systems have been the subject of research for more than 40 years. Recent advances in machine learning have led to a dramatic increase in research and the emergence of many successful approaches. Facial recognition or identification technology is more important today than any time in the past as it is very common and introduces privacy issues on how it captures and shares facial information. Latency and power consumption of these applications are important for many mobile or IoT devices in addition to the privacy issues.

This application note investigates the running of a Face Identification (FaceID) at the edge with minimized latency and optimized power consumption using a MAX78000 CNN inference engine. The challenge in this application is to design a CNN architecture with high performance while keeping the number of coefficients in the network up to 300 times less than many cutting-edge deep face identification networks, i.e., DeepFace [1].

Convolutional Neural Networks (CNN) are very useful as they allow the learning of position and scale independent features from the input data. The length of the convolution kernels is generally small, i.e., 3 x 3, 7 x 7, etc., which provides great memory efficiency. The interest for these networks has increased as many different studies show the provided performance gain as well as the reduction in the model size. The downside of such architectures is the higher computational loads that can create high energy consumption and latency. These concerns are overcome in this study by the unique design of the MAX78000.

This document briefly introduces the MAX78000 CNN inference engine and presents a methodology to develop a model to identify faces that can fit into the chip. Then it describes the synthesis of the developed model and explains the application software for the MAX78000 EV kit.

CNN Inference Engine – MAX78000

The MAX78000 [2] is a new breed of Artificial Intelligence (AI) microcontroller built to enable neural networks to execute at ultra-low power and live at the edge of the IoT. This product combines the most energy-efficient AI processing with Maxim's proven ultra-low power microcontrollers. The hardware-based CNN accelerator enables battery-powered applications to execute AI inferences while spending only microjoules of energy. This makes it an ideal architecture for energy-critical applications. The MAX78000 features an Arm® Cortex®-M4 with a Floating-Point Unit (FPU) CPU for efficient system control with an ultra-low-power deep neural network accelerator. **Figure 1**. The architecture of the MAX78000. shows the top-level architecture of the MAX78000.

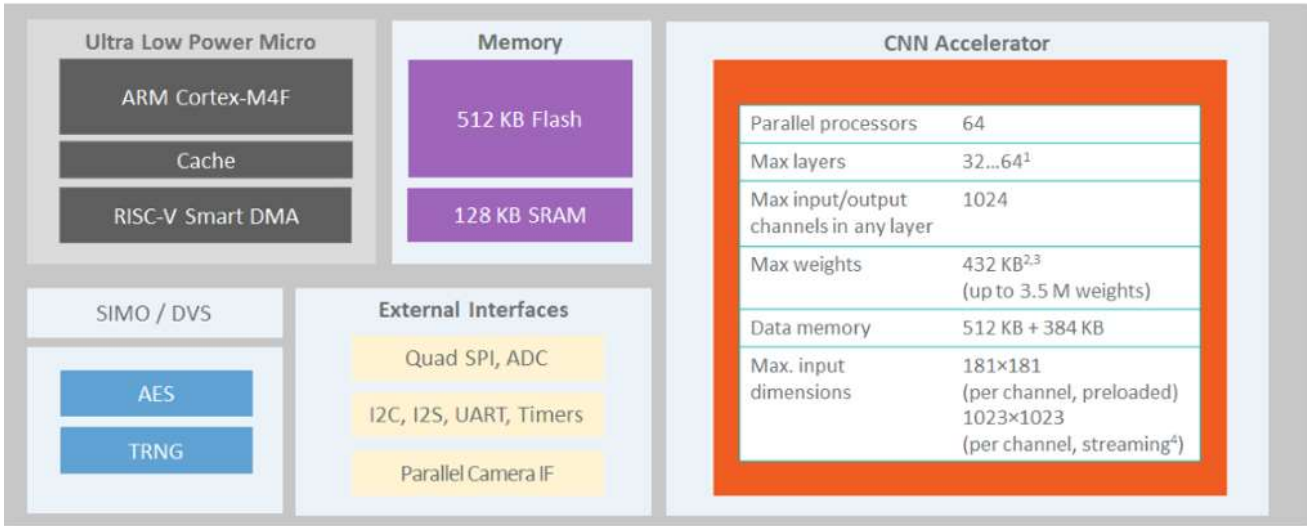


Figure 1. The architecture of the MAX78000.

The MAX78000 evaluation kit [3] provides a platform to leverage the capabilities of the MAX78000 to build new generations of AI devices. The EV kit features onboard hardware like a digital microphone, serial ports, camera module support, and a 3.5in touch-enabled color thin-film transistor (TFT) display.

MAX78000 Development Flow

The PyTorch or TensorFlow-Keras toolchain can be used to develop a model for the MAX78000. The model is created with a series of defined subclasses representing the hardware. Some operations like pooling or activations are fused to the 1D or 2D convolution layers, and fully connected layers. Rounding and clipping are also added to match the hardware.

The model is trained with floating-point weights and training data. Weights can be quantized either during training (quantization aware training) or after training (post-training quantization). The result of quantization can be evaluated over the evaluation dataset to check the accuracy degradation due to weight quantization.

The MAX78000 synthesizer tool (ai8xize) accepts PyTorch checkpoint or TensorFlow exported ONNX files as an input, as well as the model description in the YAML format. An input sample data file (.npy file) is provided to the synthesizer as well to verify the synthesized model on the hardware. The inference outcome for this data file is compared with the expected output of the pre-synthesis model.

The MAX78000 synthesizer automatically generates the C code, which can be compiled and executed on the MAX78000. The C code includes Application Programming Interface (API) calls to load the weights as well as the provided sample data to the hardware, execute an inference on the sample data, and compare the classification outcome with the expected result as a pass/fail sanity test. This generated C code can be used as an example to create custom applications. **Figure 2** shows the overall development flow of the MAX78000.

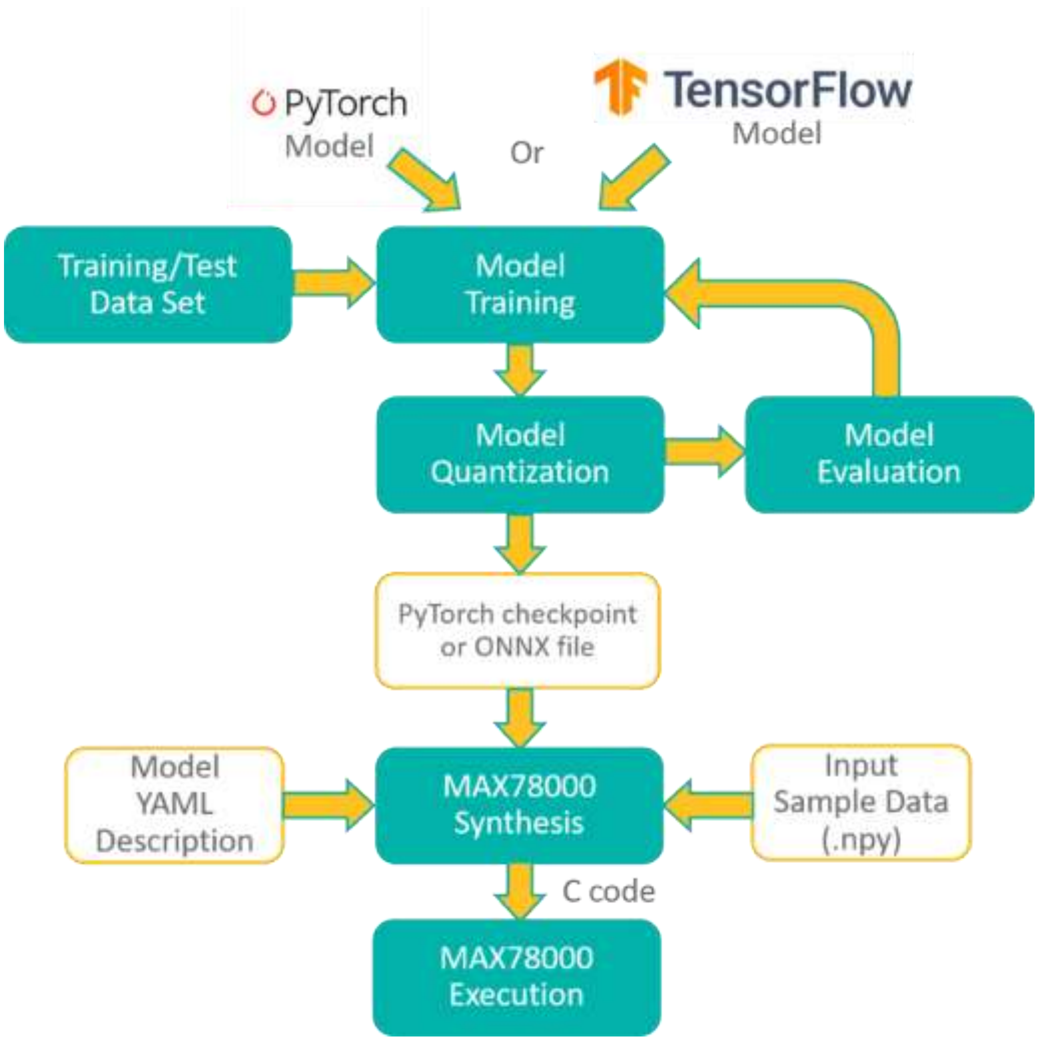


Figure 2. Development flow of the MAX78000.

FaceID Model Development Methodology

The face identification problem is solved in three main steps:

- Face Extraction: Detection of the faces in the image to extract a rectangular subimage that contains only one face.
- Face Alignment: Determination of the rotation angles (in 3D) of the face in the subimage to compensate its effect by affine transformation.
- Face Identification: Identification of the person using the extracted and aligned subimage.

There are different approaches for the first two steps. Multitask-cascaded convolutional neural networks (MTCNN) [4] solve both the face detection and alignment steps. Face identification is generally studied as a different problem, which is the focus of this demonstration. The MAX78000 EV kit is used to identify the uncropped portrait faces, each containing one face only.

The adopted approach is based on learning a signature, i.e., embedding, for each facial image whose distance to another embedding gives a measure about the face's similarity. It is expected to observe small distances between the faces of the same person and large distances for the faces of distinct people.

FaceNet [5] is one of the most popular CNN-based models developed for the embedding-based face identification approach. The triplet loss is the key behind its success. This loss function takes three input samples: the anchor, positive sample from the same identity with the anchor, and negative sample from a different identity. The triplet loss function gives low values when the distance of the anchor is close to the positive sample and distant to the negative one (**Figure 3**).

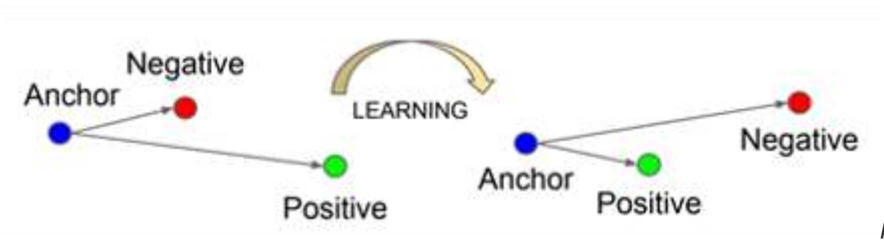


Figure 3. Triplet Loss minimizes the distance between the anchor and the positive, and maximizes distance between the anchor and the negative [2].

However, this model has 7.5 million parameters, which is too large for the MAX78000. It also requires 1.6G floating point operations, which makes the model very hard to run on many mobile or IoT devices. Therefore, a new model architecture is designed with less than 450k parameters to fit into the MAX78000.

A knowledge distillation approach is adopted to develop this tinier CNN model from FaceNet as it is a widely appreciated neural network for FaceID applications.

Knowledge distillation in machine learning is the process of transferring knowledge from a large model to a smaller one [6]. Large models have higher knowledge capacity than small models. Yet, the capacity might not be fully utilized. So, the aim here is to teach the exact behavior of the big network to the smaller network.

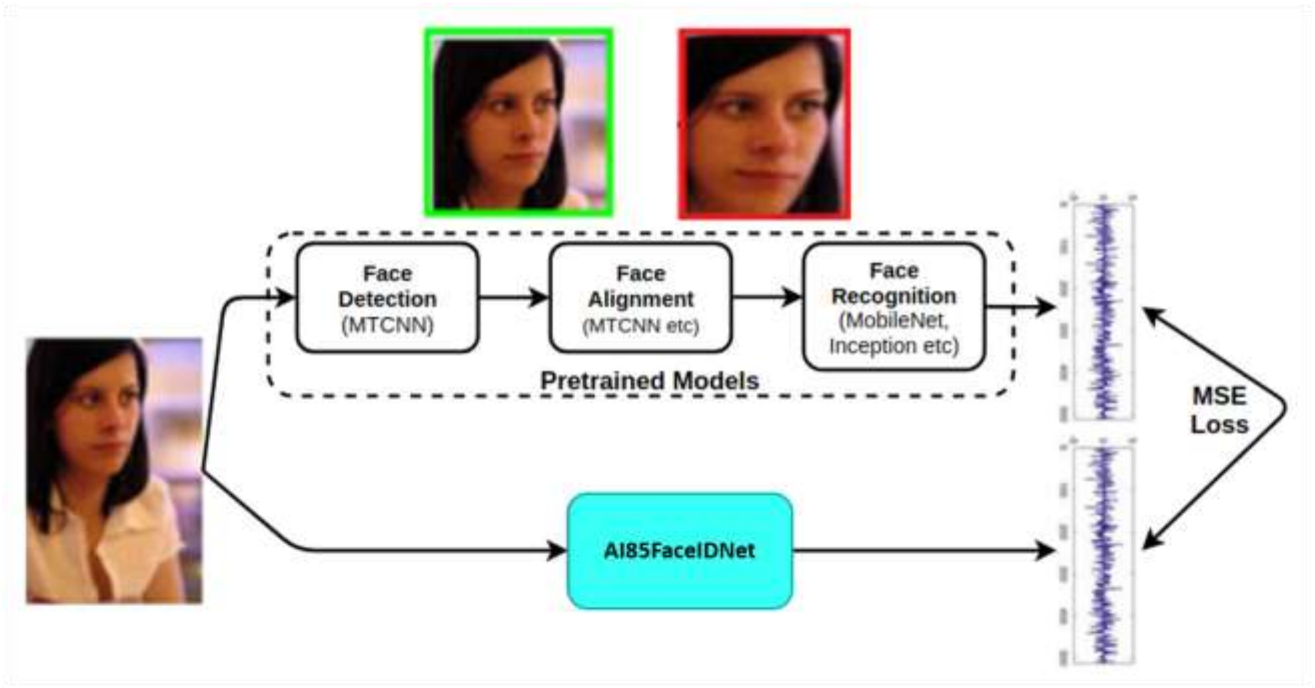

Face Detection (MTCNN)
Face Alignment (MTCNN etc)
Face Recognition (MobileNet, Inception etc)
Pretrained Models
MSE Loss
AI85FaceIDNet

Figure 4. The approach for model development.

Figure 4. The approach for model development summarizes how pretrained MTCNN and FaceNet models are utilized to develop the compact FaceID model, AI85FaceIdNet. The embeddings FaceNet model is used as the AI85FaceIdNet's target. There is no center loss, triplet loss, etc., as these are assumed to be covered by the FaceNet model. The loss used in the model development is the Mean Square Error (MSE) between the target and predicted embeddings, which is also the distance to define the face similarity.

Datasets

The training of the model is done using the following datasets:

- VGGFace-2 [7]: A large-scale face recognition dataset.
- YouTubeFaces [8]: A database of face videos designed to study the problem of unconstrained face recognition.

A dataset is created from the six images found on the web for each of the selected 15 female and 15 male celebrities (**Figure 5** and **Figure 6**) to test the model. This dataset is called MaximCeleb in the rest of this document.

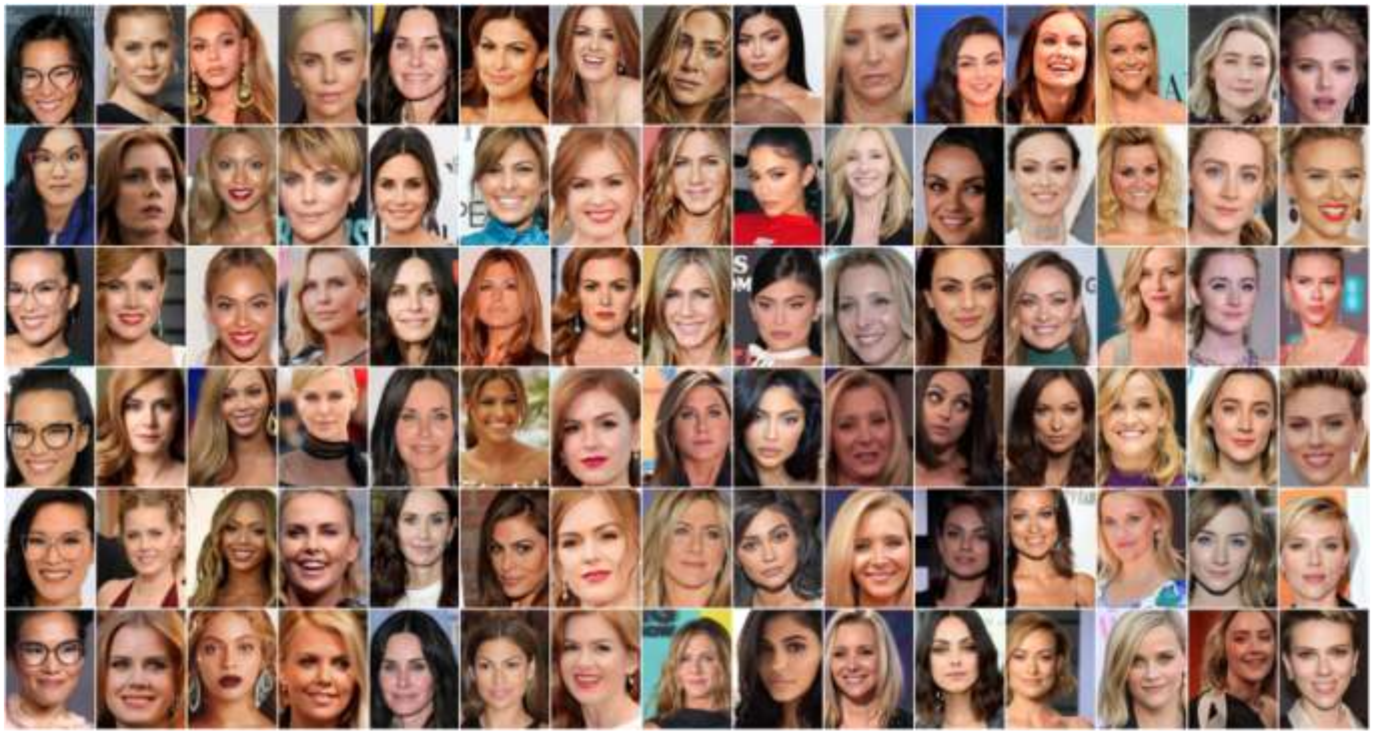


Figure 5. Images of female celebrities in the MaximCeleb dataset.

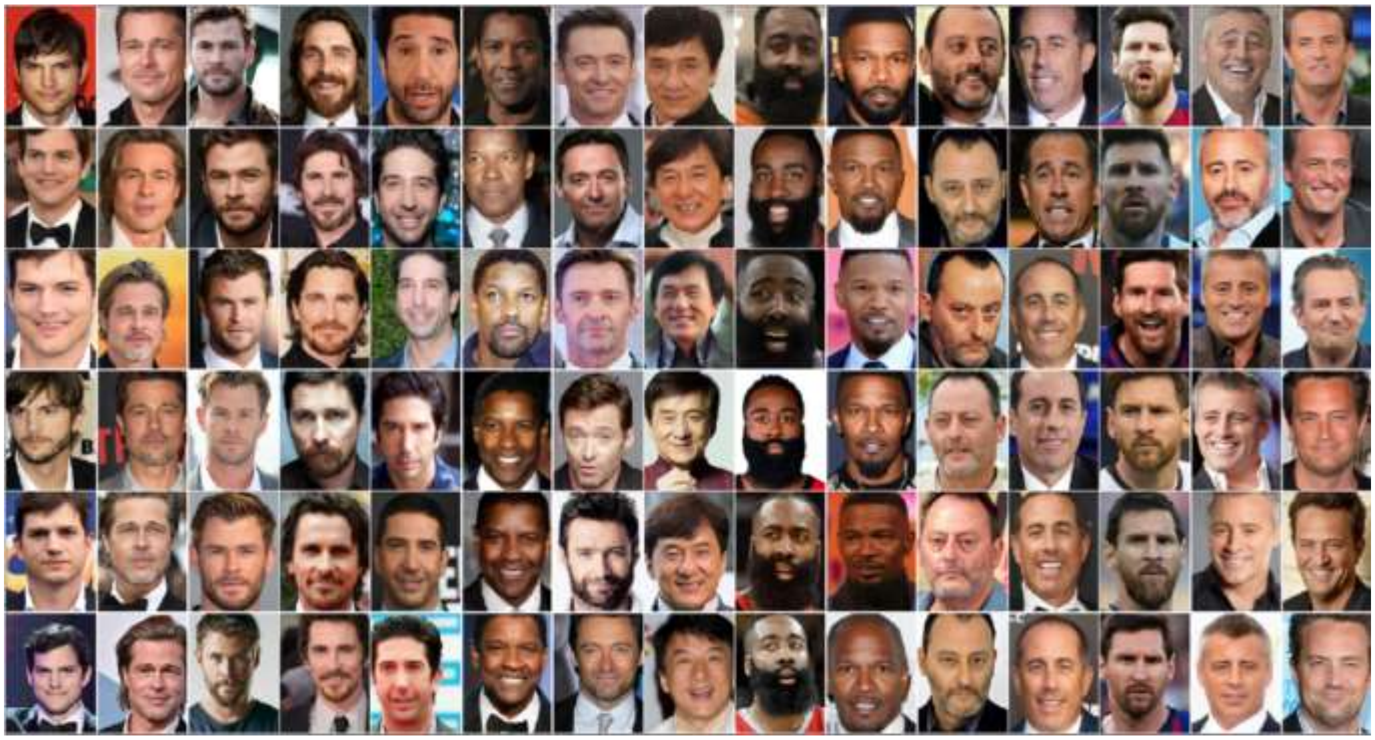


Figure 6. Images of male celebrities in the MaximCeleb dataset.

Dataset Generation and Augmentation

A 120 x 160 x 3 (width x height x depth) subimage is randomly cropped for each image in the datasets. The detected and aligned face in the subimage by a pretrained MTCNN model is fed to a pretrained FaceNet model to create embeddings. Note that both the retrained models are taken from [9]. So, the length of the resultant embeddings is 512. Finally, 120 x 160 RGB facial images are stored with its owner and embedding to be used during the training.

The location and orientation of the face in the new image varies. It is expected to have a model that is robust against small amount of transformations of the head in the image.

CNN Model Training

The MAX78000 FaceID model, AI85FaceIdNet, consists of eight sequential convolutional blocks. **Figure 7**. AI85FaceIdNet network structure shows the CNN model. Some layers include pooling operations to decrease the size of the inputs. Similarly, a 512 x 5 x 3 sized tensor is averaged with a (5 x 3) kernel to obtain 512-sized embeddings at the last layer.

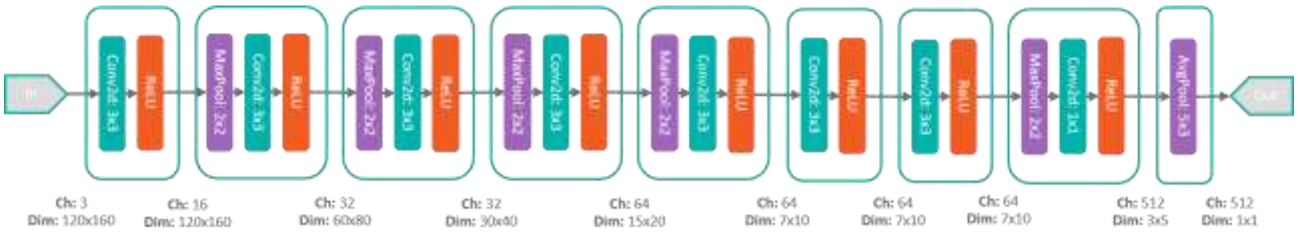


Figure 7. AI85FaceIdNet network structure.

The model is trained with Maxim tools using the following command:

```
train.py –epochs 100 –optimizer Adam –lr 0.001 –deterministic –compress schedule-faceid.yaml –model ai85faceidnet –dataset FaceID --batch-size 100 –device MAX78000 –regression
```

The script creates a checkpoint file of the model at the end of the training process with the highest score on the validation set. Then, the floating points weights are quantized as the MAX78000 is an integer arithmetic device. The conversion is also trivial with Maxim tools using the following command:

```
./quantize.py   --device MAX78000 -v -c networks/faceid.yaml –scale 1.05
```

Model Performance

The performance of the model is analyzed using the MaximCeleb dataset containing faces of 30 celebrities (15 female and 15 male).

The performance of the models is evaluated separately for female and male datasets in terms of the accuracy of identification of each image by use of the distances of its own embedding to the rest of the 89 images' embeddings. The distance metric and algorithm to define the closest embedding are also the one in the application.

The L2(Euclidian) norm is chosen as the distances between the embeddings in this analysis. The average distances of the embeddings of all the subjects to the given image's embedding is extracted to determine the closest subject. Then, the algorithm returns the subject whose embeddings are the closest in average to the given embedding. Table 1 shows the performances of the combination of the MTCNN and FaceNet, and the AI85FaceIdNet running on the MAX78000. **Figure 8** and **Figure 9** show the confusion matrices for each dataset.

Table 1. The Performances of the Female and Male MaximCeleb Subjects

	Maximceleb Dataset			
	Female		Male	
	MTCNN+FACENET	AI85FACEID	MTCNN+FACENET	AI85FACEID
ACCURACY (%)	94.4	78.9	94.4	88.9

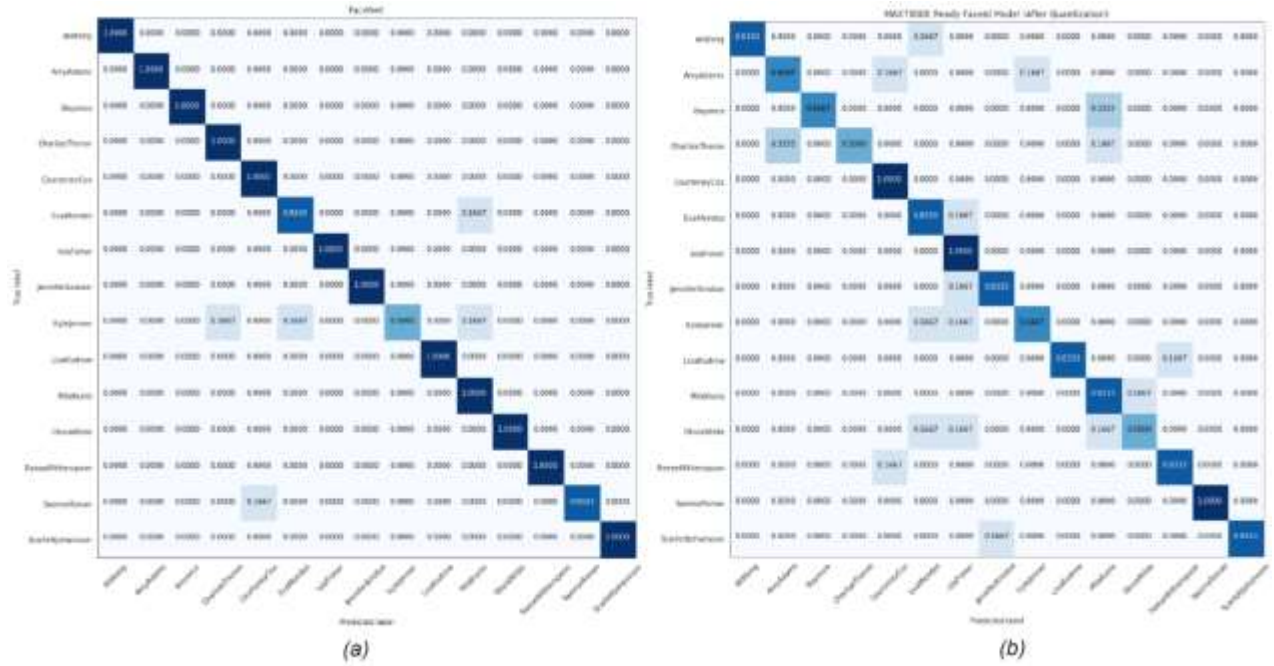


Figure 8. The confusion matrices for the (a) MTCNN+FaceNet (b) AI85FaceIdNet models for the female MaximCeleb dataset.

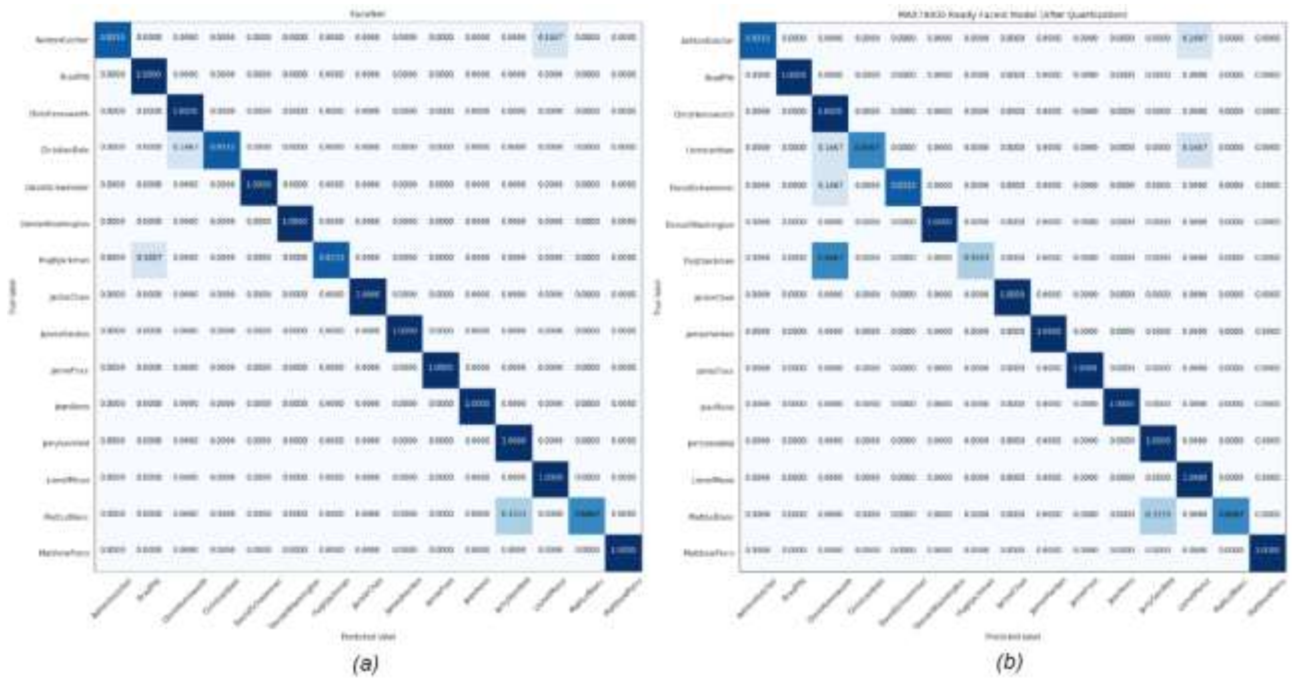


Figure 9. The confusion matrices for the (a) MTCNN+FaceNet (b) AI85FaceIdNet models for the male MaximCeleb dataset.

Identifying the Unknown Subjects

The Receiver Operating Characteristic (ROC) curve of the developed model is used to identify the unknown subjects. The ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold varies [10].

The previous section defines the distance of a face to the subject as the mean L2 distance of the face embedding to the whole embeddings stored for each subject. The decision is made upon determining the subject whose distance to the face embedding is minimum. A threshold for the minimum distance can be determined using the ROC of the model as each image is identified as one of the subjects in the database.

The ROC curve is created by plotting the true positive (TP) rate against the false positive (FP) rate at various threshold values. So, as it is moved on the ROC curve, the threshold setting is changed, and the threshold value can be determined with respect to the selected performance.

The 30-subject Maxim Celebrity dataset is used to generate the ROC curve, where only 15 female (or male) subjects are assumed in the database. The rest are determined as unknown subjects. A correctly classified sample increases the TP rate while the other samples increase the FP rate. **Figure 10** shows the ROC curves of the AI85FaceId model for the male and female datasets.

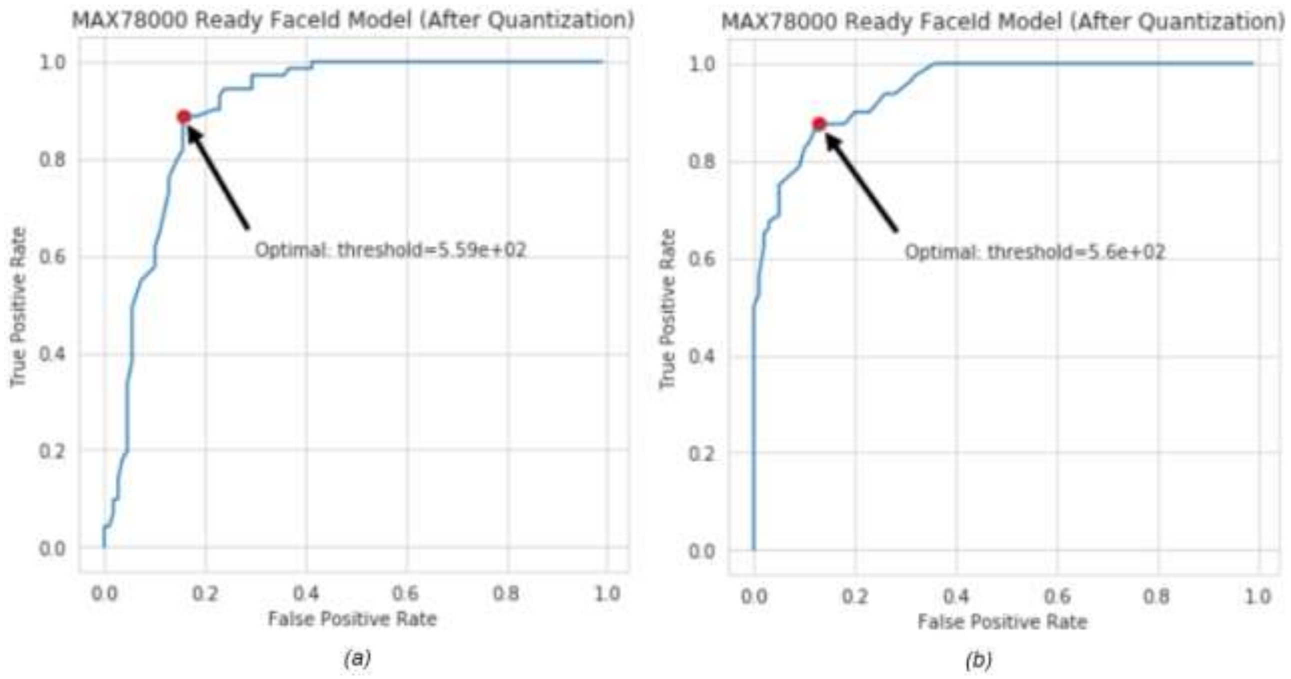


Figure 10. The ROC curves for the AI85FaceIdNet for the (a) female and (b) male MaximCeleb datasets.

There are different approaches to determine the optimum threshold value. The point on the ROC curve closest to the point (TP = 1.0, FP = 0.0) is selected as the threshold value (Figure 10). The optimum value is very close to 560 for both sets. Therefore, the unknown subject threshold is set to 560 for the FaceID application.

CNN Model Synthesis

The quantized model obtained by the steps given in the CNN Model Training section is synthesized for the MAX78000 using a python script in the Maxim tools. This script generates a C code, which includes functions to initialize the CNN accelerator, load quantized CNN weights, run the model for the given example input sample, and fetch the model output from the device once the following three items are prepared:

- Quantized PyTorch checkpoint file or TensorFlow model exported to ONNX format.
- Network model YAML description.
- A sample input with the expected result to include in the generated C code for verification.

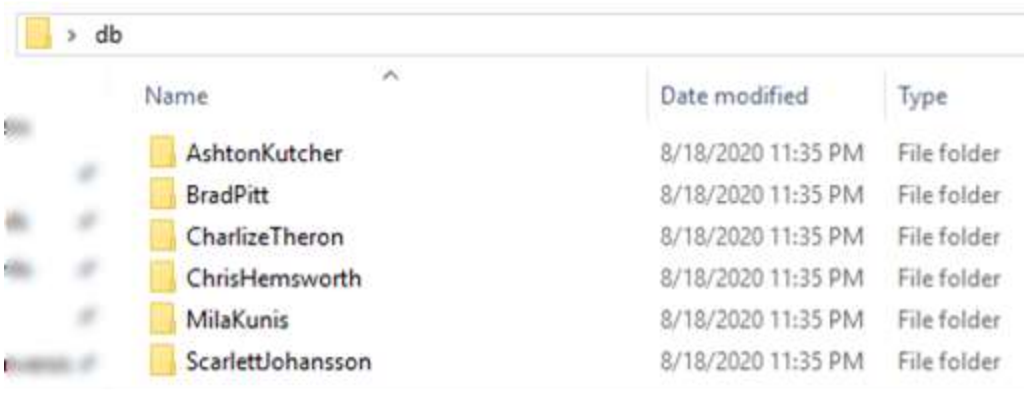
The following script is run to generate the synthesized code:

```
./ai8xize.py -e --verbose --top-level cnn -L --test-dir  --prefix faceid --checkpoint-file  --config-file networks/faceid.yaml --device M
AX78000 --fifo --compact-data --mexpress --display-checkpoint --unload
```

The barebone C code is used as the base to build the FaceID demo application to initialize the device, load the weights (kernels), push the sample input, and fetch the results.

Generating the Embedding Set

The header file that includes the embeddings of the subjects, embeddings.h, is altered to create the custom dataset to run the FaceID demo. The Python script, generate_face_db.py, in the db_gen folder is used for this purpose. The sample use of the script is given in gen_db.sh. The script takes the folder name where the images of subjects are stored as an argument. This folder must include separate subfolders for each subject and these subfolders must be named with the names or identifiers of the subjects. **Figure 11** shows the sample db folder structure. The images of the subjects are placed in the associated folders.



Name	Date modified	Type
AshtonKutcher	8/18/2020 11:35 PM	File folder
BradPitt	8/18/2020 11:35 PM	File folder
CharlizeTheron	8/18/2020 11:35 PM	File folder
ChrisHemsworth	8/18/2020 11:35 PM	File folder
MilaKunis	8/18/2020 11:35 PM	File folder
ScarlettJohansson	8/18/2020 11:35 PM	File folder

Figure 11. Sample database folder structure to generate the custom embedding set.

Calling with the arguments as in gen_db.sh and generate_face_db.py generates the embeddings.h automatically with the images and subjects in the db folder. Once the file is changed, the next build becomes a customized version with the updated embeddings list.

```
$ ./gen_db.sh
```

The face identification with the MTCNN, cropping the 120 x 160 frame with the detected face, and illumination correction is carried out in the generate_face_db.py. The images must be larger than 120 x 160 and must only include one subject facing the camera. At least five images per subject are recommended to increase the identification accuracy.

FaceID Demonstration Platform

The face recognition is demonstrated on the MAX78000 EV kit using the FaceID firmware. The EV kit (**Figure 12**. Screenshots from the FaceID application on the MAX78000 with the subject database given in Figure 11) consists of a TFT screen and a Video Graphics Array (VGA) camera, both directed upwards to operate in the selfie mode. The demo application continuously runs and reports one prediction for each frame. The unknown class is reported if the closeness of the extracted embedding to the subjects in the database does not exceed a predefined threshold. The threshold is obtained as described in the previous sections and can be updated (variable thresh_for_unknown_subject in the embedding_process.h). The L1 (Manhattan) distance can be used instead of the L2 norm as it requires less computation. But the unknown threshold determination step must be repeated accordingly.

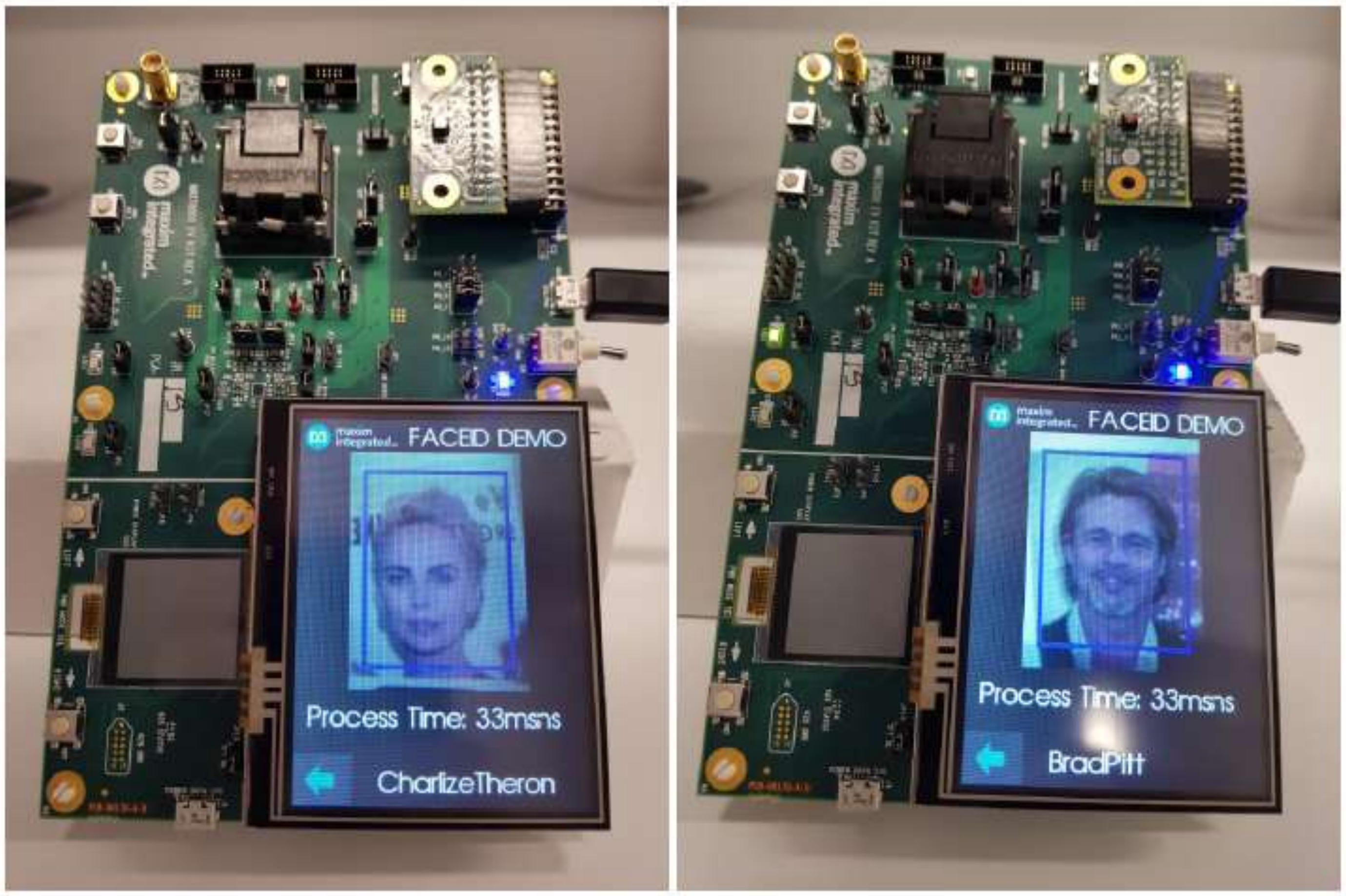


Figure 12. Screenshots from the FaceID application on the MAX78000 with the subject database given in Figure 11.

The demo on the MAX78000 EV kit runs in the portrait position as it is more suitable to shoot and display selfies. The face must be in the blue rectangle as this portion is the center of the regions to be cropped for processing. Three 120 x 160 images are cropped from the raw capture with different X and Y offsets. These are fed to the CNN model to increase the accuracy as an augmentation method. The consistency of the predictions is tested before reporting them by sending more than one image that covers the face to the CNN. Likewise, the consistency of predictions in consecutive time samples are also tested. This may introduce some lag to the identification speed but increases the robustness of the application. Figure 13 shows the flow diagram of the application.

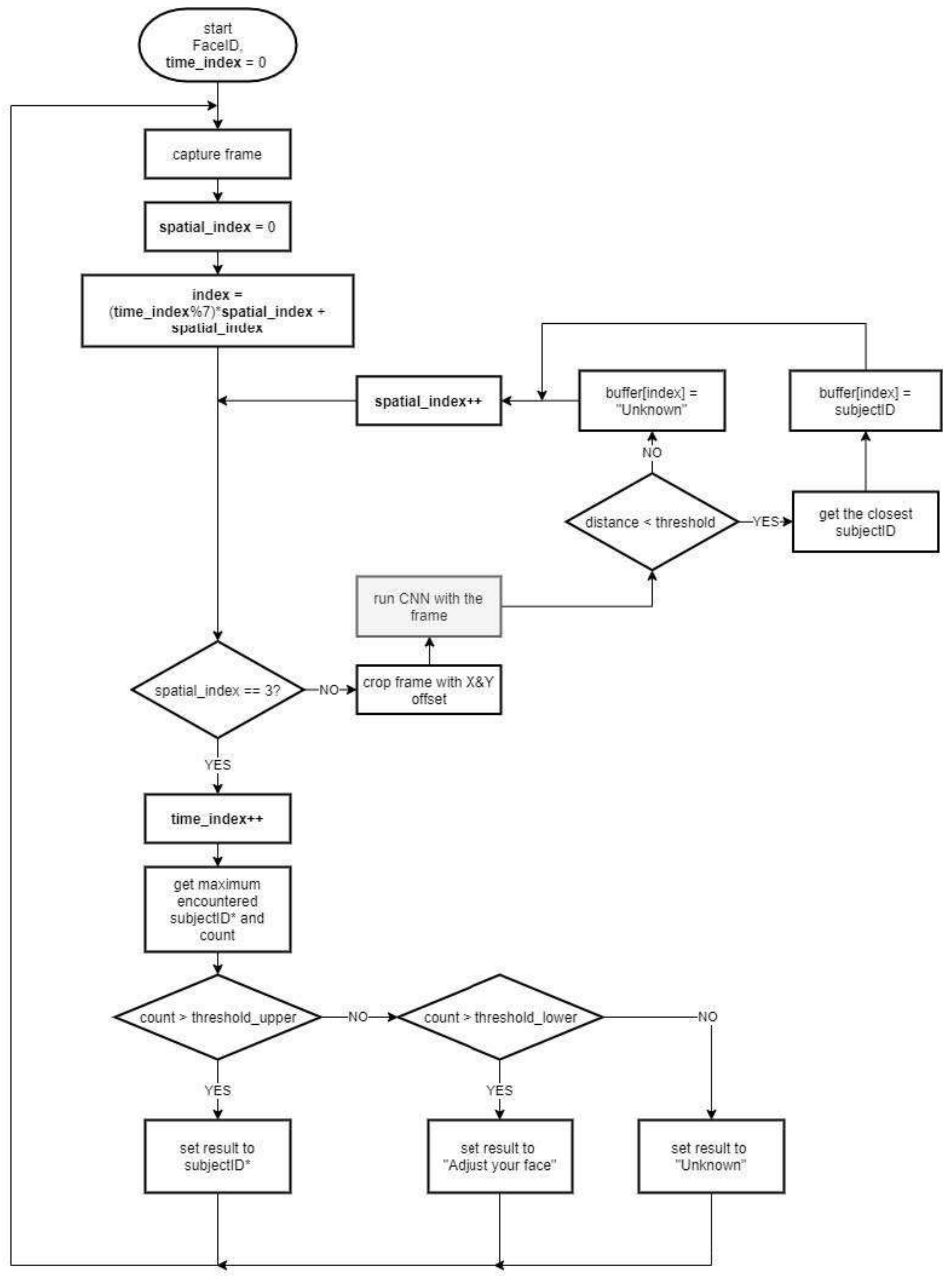


Figure 13. Process flow of the FaceID demo application on the MAX78000.

Conclusion

This application note demonstrates the implementation of a face identification model in the MAX78000 and its deployment on the ultra-low power MAX78000 platform for the resource constrained edge or IoT applications. This application follows a knowledge distillation-based model development methodology to meet the MAX78000 requirements. So, this document can also be a guide to migrate high-performing large networks to the edge devices. The application note presents the performance analysis of the model as well as the required steps to synthesize the model to deploy the MAX78000.

References

[1] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 1701-1708, doi: 10.1109/CVPR.2014.220.
[2] MAX78000 Data Sheet
[3] MAX78000 Evaluation Kit Data Sheet
[4] Zhang, Kaipeng et al. "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks." IEEE Signal Processing Letters 23.10 (2016): 1499–1503.
[5] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering." 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
[6] https://en.wikipedia.org/wiki/Knowledge_distillation
[7] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, A. Zisserman VGGFace2: A dataset for recognizing face across pose and age International Conference on Automatic Face and Gesture Recognition, 2018
[8] https://www.cs.tau.ac.il/~wolf/ytfaces/
[9] https://github.com/timesler/facenet-pytorch
[10] https://en.wikipedia.org/wiki/Receiver_operating_characteristic

Related Parts		
MAX78000	Ultra-Low-Power Arm Cortex-M4 Processor with FPU-Based Microcontroller with Convolutional Neural Network Accelerator	Free Sample
MAX78000EVKIT	Evaluation Kit for the MAX78000	