IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 May 2023, accepted 22 June 2023, date of publication 28 June 2023, date of current version 6 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3290902

RESEARCH ARTICLE

Fast Search of Face Recognition Model for a Mobile Device Based on Neural Architecture Comparator

ANDREY V. SAVCHENKO[1,2,3], LYUDMILA V. SAVCHENKO[2], AND ILYA MAKAROV[3,4]

[1]Sber AI Lab, 117312 Moscow, Russia
[2]Laboratory of Algorithms and Technologies for Network Analysis, HSE University, 603155 Nizhny Novgorod, Russia
[3]AI Center, NUST MISiS, 119049 Moscow, Russia
[4]Artificial Intelligence Research Institute (AIRI), 105064 Moscow, Russia

Corresponding author: Andrey V. Savchenko (avsavchenko@hse.ru)

The publication was supported by the grant for research centers in the field of AI provided by the Analytical Center for the Government of the Russian Federation (ACRF) in accordance with the agreement on the provision of subsidies (identifier of the agreement 000000D730321P5Q0002) and the agreement with HSE University No. 70-2021-00139.

ABSTRACT This paper addresses the face recognition task for offline mobile applications. Using AutoML techniques, a novel technological framework is proposed to develop a fast neural network-based facial feature extractor for a concrete device. First, the Once-for-All SuperNet is trained on a large facial dataset. Each device is characterized by its lookup table, which contains the running times of inference in each layer of the SuperNet. An evolutionary search is then used to select the most accurate subnetwork within a limit on the maximum expected latency. It is proposed to train a neural architecture comparator using Gradient Boosted Trees to choose the better subnetwork in this search. Experimental face verification and recognition results demonstrate the robustness of the novel method to various facial region positions. The best model achieves an identification accuracy of 98.7% for the LFW dataset in less than 5 ms on the Qualcomm Snapdragon 865 GPU.

INDEX TERMS AutoML, neural architecture search (NAS), convolutional neural network (CNN), face recognition, mobile device, once-for-All SuperNet.

I. INTRODUCTION

One of the most challenging pattern recognition problems is face verification and identification tasks [1], [2]. In typical scenarios, the training set contains a small number of photos per each subject of interest [3]. As a result, these tasks are solved nowadays by extracting features (embeddings, descriptors) with a deep neural network, pre-trained on large external facial datasets [4]. These feature vectors are classified by an arbitrary technique, such as k-NN (k-nearest neighbor).

In this paper, we are focused on mobile applications with offline facial processing. The most time-consuming part of the above-mentioned conventional procedure is the inference in a deep neural network. This part may be too computationally expensive for real-time facial processing on a mobile or edge device, e.g., in video-based face recognition. Hence, the lightweight CNN (convolutional neural network) architectures should be used [5]. Unfortunately, choosing the best architecture for a concrete mobile device is difficult. Indeed, the computational power of cheap and expensive smartphones is significantly different, so it is impossible to find a single CNN characterized by high accuracy and reasonable performance for an arbitrary device. One potential solution here is the usage of AutoML (Automated Machine Learning) and NAS (Neural Architecture Search) techniques for a proper choice of the neural network for a concrete device [6], [7]. As the traditional AutoML-based training of a specialized descriptor for a substantial device takes too much time, we will borrow the idea of the OFA (Once-for-All) SuperNet [8] that is trained only once. Still, the specialized CNNs can be rapidly extracted for a latency constraint.

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

The **objective** of this paper is to rapidly obtain a facial extraction neural network with high performance on a given mobile device without the need to re-train the model. Here are our main **contributions**:

1) a novel technological framework for efficient NAS to extract specialized CNN-based facial features under hardware-aware constraints;
2) the novel GBDT (gradient boosting decision tree)-based NAC (neural architecture comparator) and the evolutionary search using the QuickSelect algorithm. Our experimental study demonstrates the benefits of the proposed approach over the original MLP (Multi-Layered Perceptron)-based accuracy predictor from OFA [8];
3) We obtain several models with low latency by using our framework and making them publicly available.

The remaining part of the paper is structured as follows. In Section II, face recognition and AutoML techniques are reviewed. The problem statement for face recognition is presented in Section III. In Section IV, we show the details of the proposed approach. Section V describes the LFW (Labeled Faces in the Wild) dataset [9] and known facial descriptors to be compared with our models. Section VI presents experimental results and ablation study of obtained neural networks. Conclusion and future works are discussed in Section VII.

II. LITERATURE SURVEY

A. FACE RECOGNITION

Face recognition has been one of the most widely studied problems in pattern recognition and computer vision for more than 50 years [10]. The incredible progress in this area in the last decade is explained by the appearance of massive publicly available datasets of facial photographs [4] gathered in unconstrained environments [11] and the development of deep learning-based techniques [12]. Their main idea is to train a CNN to identify subjects from a large external dataset of celebrities and remove the last classification (fully connected) layer. The resulting neural network can be used as a feature extractor to represent the probe and gallery images as a high-dimensional facial descriptor that ideally can be classified nearly as well as if a rich dataset of photos of these individuals were present. Though the high accuracy is achieved even by using conventional softmax (categorical cross-entropy) loss function, many regularization techniques, such as ArcFace (Additive Angular Margin Loss) [1] and MagFace (Magnitude-aware loss) [13], have been proposed to improve the quality of facial representations (embeddings). In addition, special triplet loss has been actively studied since its introduction in FaceNet [2], which tries to minimize the distance between photos of the same subject and maximize the distance between different persons. The application of Lie algebra theory was examined in [14] to deal with face rotation for accurate facial feature extraction using CNNs. Moreover, adding modalities via speech recognition [15], [16], [17] or depth estimation [18], [19], [20] followed by depth refinement [21], [22], [23], [24], [25] can improve the accuracy for video-based person identification. As a result of these efforts, a great number of various applications have recently appeared in video surveillance and biometrics [26], [27].

However, many challenging issues in face recognition still need to be resolved. For example, even contemporary facial descriptors are characterized by racial bias [28], low accuracy for low- illumination images, and re-identification tasks [29]. Data augmentations for low-resolution facial images are studied in [30]. Due to the COVID-19 pandemic, masked face recognition has also been widely studied [31]. Finally, facial processing on mobile and embedded systems has been recently examined [32], [33]. The lightweight and mobile models [5] have shown high accuracy even in low-resource scenarios. To reduce the required computational cost of the existing face recognition models, the QuantFace based on low-bit precision format model quantization was applied [34]. However, it is known that mobile devices have very different computational power. Hence, it is practically impossible to train a universal facial descriptor that can be used for real-time face recognition with high accuracy on all devices. Let us consider the possibility of applying modern AutoML techniques to search the CNN that ideally fits the capabilities of a concrete device.

B. AutoML AND SuperNets FOR MOBILE DEVICES

Face recognition can be considered as a particular case of image recognition tasks. Though the loss functions are usually different, the architectures of neural networks for both tasks are generally identical. Hence, several applications of AutoML for image recognition on mobile devices are discussed in this Subsection. One of the first techniques, namely, NetAdapt (Neural Network Adaptation) [35], suggests the usage of a latency LUT (look-up table) for efficient hardware-aware NAS. The development of these ideas in the MnasNet (mobile neural architecture search) [6] finds CNN architectures with high accuracy and low latency based on a customized weighted product. Unlike previous works, where latency is considered via an inaccurate proxy (e.g., FLOPS, FLoating-point OPerations per Second), the MnasNet directly measures the latency by executing the inference on a mobile phone. The combination of MobileNetV2 with Squeeze-and-Excite blocks, swish activation functions, and Platform-aware NAS with NetAdapt caused the development of MobileNetV3 [36], which was the first network from the MobileNet family that was obtained using AutoML. However, the most valuable result was uniformly scaling depth/width/resolution using a simple yet highly effective compound coefficient. It was integrated into AutoML to obtain the state-of-the-art EfficientNets [37]. Moreover, efficient neural networks without the need to evaluate candidate models can be obtained by the Differentiable Neural Architecture Learning based on architecture parameterization based on scaled sigmoid function [7]. Automatic search space and search strategies regarding combinatorial optimization in the Layered Architecture Search Tree [38] have been considered.

Based on weight-sharing techniques, one-shot architectures (SuperNets) for AutoML were suggested in [39]. The Single Path One-Shot [40] uses a SuperNet with shared parameters for efficient training. The Contrastive Neural Architecture Search [41] estimates the performance of candidate architectures by computing the probability of candidates being better than a baseline one using graph convolutional network-based NAC. The novel shrinking strategy that progressively simplifies the original search space by discarding unpromising operators for SuperNets was proposed in [42]. Hardware constraints for one-shot NAS that do not require finding a specialized neural network and training it from scratch for each device were introduced in the OFA SuperNet [8]. It supports diverse architectural settings by decoupling training and searches to reduce the cost of getting a specialized subnetwork from the OFA SuperNet without additional training. Its core component is the evolutionary search for the best architecture with suitable latency using the MLP-based accuracy predictor. The latter solves a simple regression task and tries to predict the validation accuracy based on a description of a subnetwork. An extended version of the OFA was proposed in APQ (Architecture, Pruning, and Quantization) [43], which implements the joint search for network architecture, pruning, and quantization policy by training a surrogate predictor for pruning and quantization-aware scheme.

Though most AutoML techniques have been used in image recognition tasks, several papers apply NAS to train facial descriptors. For example, the differential NAS architecture has been implemented in the highly lightweight PocketNet with high performance [44]. Various AutoML methods have been developed for the loss function search in face recognition [45] and person re-identification [46]. However, all such articles introduce efficient neural architectures that cannot be adapted to generate face recognition networks for a concrete device rapidly. Our paper fills this gap using the OFA-based SuperNet with device-specific facial feature extraction.

III. FACE RECOGNITION BASED ON DEEP EMBEDDINGS

A. FACE IDENTIFICATION

In the face identification task, it is required to assign an observed facial image X to one of $C \geq 1$ classes of subjects (identities) or make a decision that an observed person does not belong to the list of known subjects (open-set scenario). These subjects are specified by the gallery set of $N \geq C$ facial images with known subject identifier $c(n) \in \{1, \ldots, C\}$ of the n-th photo ($n = 1, 2, \ldots, N$).

Due to the complexity of gathering many facial photos of the subjects of interest, the training set is typically very small ($C \approx N$) for training a complex classifier. Hence, domain adaptation and feature learning are commonly applied. At first, a deep neural network is pre-trained for face identification from large datasets of celebrities, e.g., CASIA-WebFace (Chinese Academy of Sciences Institute of Automation), MS-Celeb-1M (Microsoft Celebrities with 1 Million photos) or VGGFace/VGGFace2 (Visual Geometry Group Facial datasets) [4]. Next, the last classification layer is removed. Every n-th training example is fed into this CNN to be described as a descriptor $\mathbf{x}_n = [x_{n;1}, \ldots, x_{n;D}]$ at the output of the penultimate layer. Its dimensionality D is relatively high: it typically varies in a range $[512, 4096]$. The input facial image is associated with embeddings $\mathbf{x} = [x_1, \ldots, x_D]$ using the same procedure. Finally, an arbitrary classifier, such as k-NN, is applied to these descriptors to solve the original task:

$$c^* = \underset{c \in \{1,\ldots,C\}}{\operatorname{argmin}} \rho_c(\mathbf{x}), \tag{1}$$

where

$$\rho_c(\mathbf{x}) = \min_{r \in \{1,\ldots,N\}, c(n)=c} \rho(\mathbf{x}, \mathbf{x}_n). \tag{2}$$

If the feature vectors are normalized in the L_2 norm, the Euclidean (L_2) metric can be used as the dissimilarity measure $\rho(\mathbf{x}, \mathbf{x}_n)$. Indeed, it is equivalent to the conventional cosine distance between unnormalized descriptors for the 1-NN rule (1).

In this paper, we are apprehensive about the performance of face recognition on mobile or edge devices. The feature vectors of the training examples are obtained at the preliminary stage. Hence, only one inference in a neural network is required for each input image during offline face recognition. To compare the efficiency of various techniques, the following multi-criteria objective function can be used [47]:

$$\max \overline{\mathrm{A}}, \quad \bar{t} \leq t_0. \tag{3}$$

Here, it is required that the average recognition time $\bar{t}$ measured on the target device is not greater than a fixed threshold t_0. The best face recognition method is characterized by maximal quality metric $\overline{\mathrm{A}}$, which can be estimated using the test set with known class labels. As classes are usually balanced in face identification, the quality measure $\overline{\mathrm{A}}$ is chosen as classification accuracy, computed as the number of correct predictions divided by the total size of the test set.

B. FACE VERIFICATION

Face verification is comparing a candidate's face to another and verifying whether it is a match. It can be considered a special case of the open-set face recognition with $C = 1$ subject so that a binary classification problem should be solved to decide if the person at the input (probe) and gallery photos are the same. The decision-making is similar to the above-mentioned procedure, extracting features using pre-trained CNN and their classification. Still, the minimal distance is compared with a predefined threshold ρ_0. If $\rho_{c^*}(\mathbf{x}) < \rho_0$, the decision is made in favor of subject c^*. Otherwise, the decision is delayed or rejected. In the simplest scenario of face verification, two facial photos, X_1 and X_2, are matched. If $\rho(\mathbf{x}_1, \mathbf{x}_2) < \rho_0$, then these images will be considered to contain photos of the same person. Otherwise, the decision will be made that faces are not matched.

The quality of the binary classification problem in criterion (3) can be estimated using different metrics. In face verification, one of the following performance measures is typically used as $\overline{A}$:

- Accuracy of face verification, i.e., the relative number of pairs correctly classified, while the threshold ρ_0 was chosen as good as possible;
- Validation rate@FAR≤0.001 (from now on "Val@1e-3"), that indicates how many image pairs of the same subjects are predicted correctly while keeping FAR (false alarm rate, probability that two images of different identities are the same) equal to 0.001;
- AUC-ROC (Area Under Curve Receiver Operating Characteristic) computed based on the distance between L_2-normed features without the need for estimation of a threshold;
- EER (Equal Error Rate), i.e., FAR for the threshold ρ_0 estimated when FAR equals FRR (false rejection rate).

IV. PROPOSED METHODOLOGY

A. ONCE-FOR-ALL SuperNet FOR FACE RECOGNITION

In this paper, the OFA NAS framework [8] is used as it ideally fits the problem of searching best architectures for specialized resource constraint deployment scenarios. At first, its authors train an extensive neural network typically using knowledge distillation with an arbitrary teacher model already trained on the same dataset (ImageNet in [8]). The training procedure combines two loss terms using the soft labels given by the teacher network and the actual labels. Next, the resulting large network's elastic version is fine-tuned using the special progressive shrinking algorithm. It samples subnetworks of smaller size with a progressively smaller resolution of the input image, then kernels, then lower depth, then width (while still sampling larger networks occasionally, as it reads). Thirdly, the prediction-based NAS method (MLP with several hidden layers) is learned in the performance/inference prediction module, from which the good subnetworks corresponding to a particular scenario are obtained. This results in a network from which one can directly extract sub-architectures for various resource constraints (latency, memory, etc.) without retraining.

As our paper mainly focuses on mobile devices, we dealt with the OFAMobileNetV3 that uses the same architecture space as MobileNetV3 [36]. Here, the subnetwork contains five groups of blocks. Each group consists of d blocks ($d \in \{2, 3, 4\}$). Each block is a dynamically changing convolutional layer with kernel size $ks \in \{3, 5, 7\}$ and several filters proportional to the scaling factor $e \in \{3, 4, 6\}$.

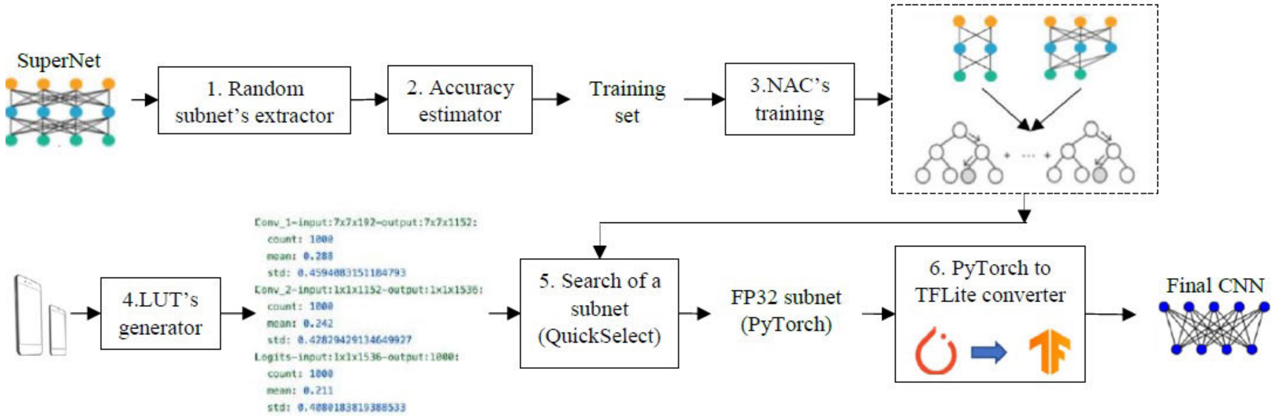
The original version of OFA [8] was developed for the general image classification task. In face recognition, it is more important for a model to obtain better facial representations rather than maximize celebrity recognition accuracy in the pre-training phase. Hence, in this paper, while using the unmodified OFA architecture, its training procedure has been changed as follows. The OFAMobileNetV3 was initially trained to recognize celebrities from a large facial dataset by implementing the progressive shrinking from the OFA repository with simple replacements of the ImageNet-related parameters, e.g., the number of output neurons, to the appropriate parameters of the new dataset. Our experiments exploited the VGGFace2 large-scale dataset with 9131 subjects [4] for this purpose. Together with the MS-Celeb-1M, the VGGFace2 is widely used to train deep facial embeddings nowadays [48]. The main advantage of training over VGGFace2 is better performance over two critical problems, pose and age variance [4]. Thus, we chose this dataset to train SuperNet in this study. The training set contained 3,067,564 photos of 9131 subjects, while the remaining 243,722 images were included in the validation set.

To account for the specifics of face recognition, we first utilized the softmax categorical cross-entropy with label smoothing and ArcFace [1] regularization. It was minimized by the SGD (stochastic gradient descent) to improve the quality of resulting facial descriptors. Secondly, as all faces have more or less the same size and shape, we simplified the training pipeline by removing support for different resolutions of the input image. Finally, as the very deep neural network training requires the teacher model, we used the EfficientNet-B0 trained on the same train-test split of the VGGFace2 dataset from our previous paper [49]. Moreover, several other tricks were employed, such as knowledge distillation with block-wise loss to train the SuperNet that most accurately corresponds to the teacher. However, we did not get an increase in accuracy compared to the basic OFA network. Hence, it was decided to leave the remaining training procedure of the OFA as close to the original version [8] as possible. The most important contribution of this paper covered in the following Subsection is the novel approach for sampling subnetworks.

B. PROPOSED FRAMEWORK

The original evolutionary search from OFA [8] used the MLP accuracy predictor to obtain the expected accuracy of a subnetwork. Training such an MLP regression model is complex because it usually overestimates the predicted accuracy. Hence, we propose modifying the evolutionary search using the NAC [41]. Let us consider the details of our novel technological framework (Fig. 1).

Its most important goal is to generate the subnetwork under specific hardware and latency constraints. At first, the dataset to train the accuracy predictor and NAC are created. The "Random subnet's extractor" unit is used to generate 16,000 random subnetworks (subnets) with various latency and accuracy using the trained SuperNet. As the resolution of the input image is fixed to 224 × 224, each subnetwork is represented as a concatenation of the following parameters: the number of layers d of each of five blocks and kernel size ks and scaling factor e for each layer. It is a goal of a search



```
Conv_1-input:7x7x192-output:7x7x1152:
  count: 1000
  mean: 0.288
  std: 0.4594083151184793
Conv_2-input:1x1x1152-output:1x1x1536:
  count: 1000
  mean: 0.242
  std: 0.42829429134649927
Logits-input:1x1x1536-output:1000:
  count: 1000
  mean: 0.211
  std: 0.4080183819388533
```

FIGURE 1. The proposed approach for fast adaptation of face recognition models.

procedure (unit 5 in Fig. 1) to find the best hyperparameters in terms of criterion (3).

Secondly, the accuracy of the resulting subnetworks on the validation part from VGGFace2 is evaluated in the "Accuracy estimator" unit to create a training dataset of pairs (subnetwork and its accuracy). Thirdly, the resulting set is used to train an architecture comparator in the "NAC's training" unit. The comparator is a binary classifier with representations of two subnetworks at the input and tries to predict if the first subnetwork is more accurate than the second one. This paper uses the GBDT from the LightGBM (Light Gradient Boosting Machine) library. The training set for this classifier is dynamically created by a 20,000-times random selection of two subnetworks from the dataset at the output of unit 2. The desired output is equal to 1 if the first subnetwork has higher accuracy than the second one or 0 otherwise. We do not use hard-negative or semi-hard negative sampling because the validation accuracy of even very small subnetwork is relatively high, so practically all training examples can be considered semi-hard.

Fourthly, the "LUT generator" unit from a particular Android application is used to create a raw version of latency LUTs to measure the running time of each possible layer of the SuperNet on a CPU (Central Processing Unit) of a specific mobile device. As it is impossible to reliably estimate the GPU (Graphical Processing Unit) latency of the whole neural network by summarizing the latency of each layer, the measurements are performed for inference on the CPU of a mobile phone. This paper computed the latency LUTs for two mobile devices with the Qualcomm Snapdragon 865 and 765 SoC (system on a chip) semiconductors.

The fifth step is the most important one. Given the LUT of the target hardware, the trained NAC, and the maximal latency constraint, the unit "Search of the subnet (QuickSelect)" performs the evolutionary search to choose the subnetwork with maximal expected accuracy, for which the running time t estimated using the LUT is not greater than the required latency t_0. The details of this step are summarized in Algorithm 1. It uses an implementation of the known QuickSelect partition algorithm based on ideas of quick sort. It re-orders the input list and efficiently returns the top-k subnetworks.

The proposed algorithm extracts the most accurate subnetworks from the current generation at each step of the evolutionary search. It contains the following hyperparameters:

1) The number T of iterations, i.e., how many generations of the population to be searched;
2) The size P of population in each generation;
3) The ratio K_r of subnetworks that are used as parents for the next generation;
4) The ratio M_r of subnetworks generated by a mutation in one generation. The remaining $P - \lfloor P \cdot M_r \rfloor$ subnetworks are chosen by crossover;
5) The probability P_m of mutation in evolutionary search.

Their default values were chosen to be identical to the values from the original evolutionary search procedure [8], namely: $T = 500, P = 100, K_r = 0.25, M_r = 0.5, P_m = 0.1$.

As the comparison of each pair of subnetworks in a generation has quadratic complexity, we implemented the partial sorting using the QuickSelect algorithm with linear complexity to split the whole generation into top-k and the remaining subnetworks. Hence, our algorithm has linear complexity depending on the number of iterations T and population size P. Searching for a concrete model requires 5–10 minutes on a GPU server.

As a result, a PyTorch subnetwork with FP32 (single-precision binary floating-point) weights is obtained. We remove the last classification layer because this CNN will be used as a feature extractor. It is converted to the TensorFlow (TF) Lite format in the "PyTorch to TFLite (TensorFlow Lite) converter" unit to be used on a mobile device. We implemented our generation of the same architecture. We copied the weights from PyTorch to TensorFlow format because the general ONNX (Open Neural Network Exchange)-based conversion led to very slow models. The resulting subnetwork is used for extraction of $D = 1536$-dimensional descriptor of facial images on a concrete mobile device. The remaining face

Algorithm 1 The Evolutionary Search of an Optimal Subnetwork

```
Initialize population of subnetworks $S := []$
Compute the number of parents $K := \lfloor P \cdot K_r \rfloor$
Compute the number of mutations $M := \lfloor P \cdot M_r \rfloor$
while size $|S|$ of the list $S$ is less than $P$ do    ▷ Get initial population of size $P$
    Randomly sample subnetwork $SN$ from the SuperNet
    Compute the latency $\hat{t}$ of $SN$ as a sum of the latencies of its layers from LUT
    if $\hat{t} < t_0$ then
        Append $SN$ to $S$
    end if
end while
for $t \in \{1, \dots, T\}$ do    ▷ Evolutionary search
    $SN_K := QuickSelect(SN, K, GBDT\ comparator)$    ▷ Get top-$K$ subnetworks (parents) from the list of subnetworks $SN$ using the GBDT comparator
    Assign first $K$ elements from $SN_K$ to $SN$
    Randomly choose $M$ subnetworks from $SN$, perform their mutation to fill the list $MSN$
    for $i \in \{1, \dots, K - M\}$ do
        Randomly choose two subnetworks from $SN$, perform their crossover and append the result to $MSN$
    end for
    $SN := MSN$
end for
Obtain top-1 subnetwork $SN^* := QuickSelect(SN, 1, GBDT\ comparator)[0]$
Remove the last classification layer from $SN^*$
return $SN^*$
```

identification/verification procedure is implemented identically to the traditional approach (Section III).

Thus, the proposed methodology (Fig. 1) is divided into three phases. During the first training phase, the SuperNet is trained on a powerful GPU server to recognize faces from a large dataset of celebrities [4]. Next, various subnetworks are randomly extracted from this SuperNet, and the accuracy of each subnetwork is estimated using the validation part of the same dataset of celebrities. Finally, the GBDT-based architecture comparator *GBDT comparator* is trained using a description of a subnetwork and corresponding validation accuracy.

The LUT is estimated for a concrete mobile device during the second deployment phase. Next, an appropriate subnetwork is extracted using the proposed Algorithm 1. Finally, during the third face recognition step, the input facial photo and every n-th example image of available subjects are preprocessed, the facial region is detected and fed into the deployed subnetwork to extract embeddings $\mathbf{x}$ and $\mathbf{x}_n$, respectively. The classifier is trained on a set of vectors $\{\mathbf{x}_n\}$, and the input facial descriptor $\mathbf{x}$ is classified to make a final decision.

C. MOBILE APPLICATION

An Android demo application was implemented to demonstrate the efficiency of the proposed approach (Fig. 1). It supports two essential functions. At first, it is possible to measure the mean and standard deviation of the inference time by 100 times running of a CNN selected in the top combo-box given a randomly initialized input tensor. Secondly, a simple face verification protocol is implemented as follows (Fig. 2). A user can select two photos from a gallery on a mobile device. Next, the facial regions are detected on each image, and the chosen subnetwork extracts the descriptors. Finally, red lines are drawn between the closed faces, for which the distance between their descriptors is less than the predefined threshold.

Our application is distributed with several models for two mobile phones with Qualcomm Snapdragon 865 and 765 chipsets. We decided to choose time constraints based on the time t_{ENet} of running the EfficientNet-B0 (TFLite) model, which is equal to 22 ± 2 ms and 59 ± 5 ms for the CPU of Snapdragon 865 and 765, respectively. To demonstrate the potential of our approach, we obtained two subnetworks (from now on, "Subnet 1" and "Subnet 2") for each device (865 and 765) by choosing two different values of t_0 (3) to be equal to 40% and 60% of the EfficientNet's inference time. The source code of the mobile application, and Jupyter Notebook to reproduce the main experiments, checkpoints of facial SuperNet, and our four subnetworks are publicly available.[1]

V. EXPERIMENTAL SETUP

A. DATA

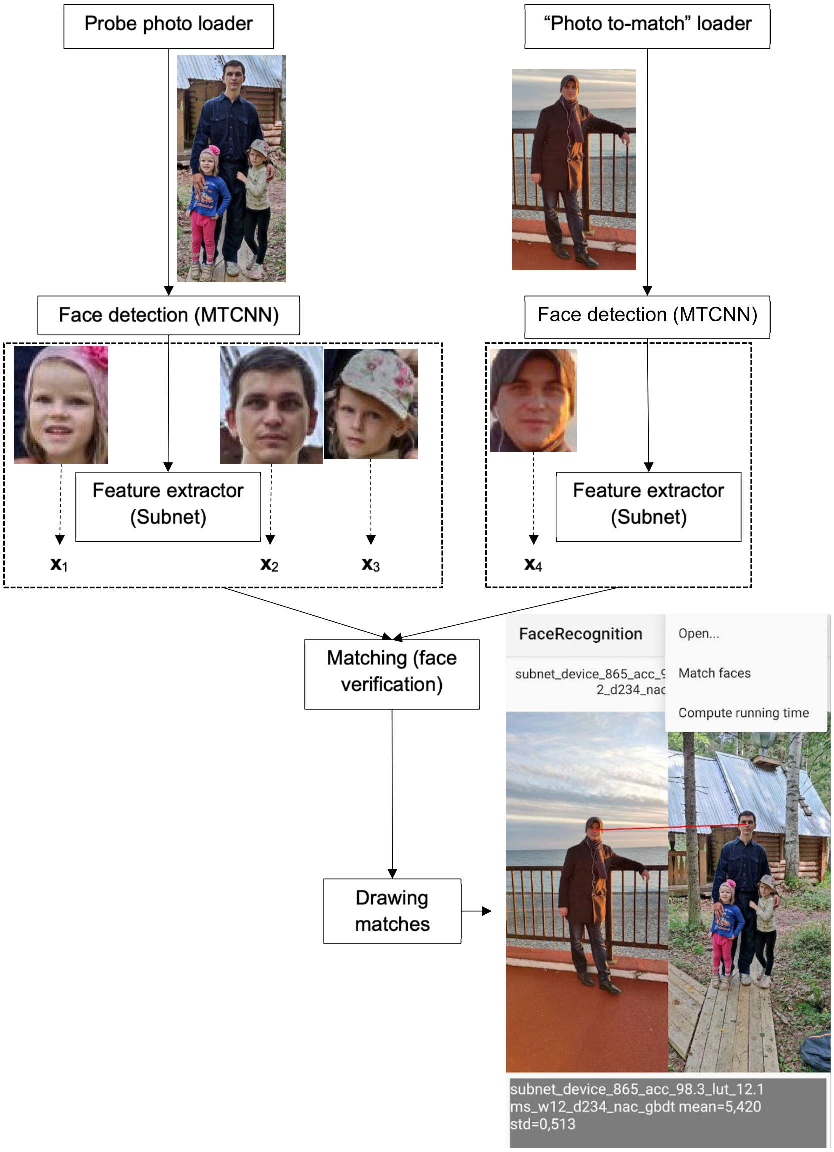
In this section, we evaluate our facial SuperNet and its four subnetworks on the LFW [9] dataset with 13233 images of 5749 people. This dataset is a de facto standard for testing face recognition algorithms [48]. All models should be assessed on this widespread public face dataset [30], [50]. We used two post-processing techniques of the faces detected by the RetinaFace [51]:

1) Simple crop of detected faces without any margins (Fig. 3a);
2) Additional face alignment from InsightFace repository using five key points at the output of the RetinaFace. Here, the similarity transform was applied, and a 224 × 224 image was obtained with some background (Fig. 3b).

Conventional evaluation protocols were used to compare the performance of models on face verification and identification tasks. In the former case, the metrics from Section III, namely, accuracy, validation rate@FAR≤0.001, AUC, and EER, were estimated using 10-fold cross-validation with splits provided by the authors of the LFW dataset.

In the latter case, classification accuracy $\overline{A}$ was estimated using the protocol from [11]: we select $C = 596$ subjects who

[1] https://github.com/HSE-asavchenko/mobile-face-recognition



subnet_device_865_acc_98.3_lut_12.1
ms_w12_d234_nac_gbdt mean=5,420
std=0,513

FIGURE 2. The data flow in the demo mobile application.

have at least two images in the LFW and at least one video in the YouTube Faces database. The training set contains precisely one facial photo of these subjects; all other images from LFW were put into the testing set. The average accuracy of the 1-NN classifier is computed using five times randomly repeated cross-validation.

B. FACIAL DESCRIPTORS

Performance of "Subnet 1" and "Subnet 2" for Snapdragon 865 and 765 is compared with several publicly-available facial descriptors pre-trained on the same VGGFace2 dataset, namely:

- IResNet-50 (Improved Residual Network, vgg2_r50_pfc ONNX model)[2] from InsightFace (ArcFace) [1];
- PyTorch version of the SENet-50 (model "senet50_ft")[3] provided by the authors of the VGGFace2 [4];
- InceptionResNet v1 from FaceNet [2] repository based on TensorFlow.[4]
- Multi-task MobileNet v1 (age_gender_tf2_224_deep-03-0.13-0.97)[5] with simultaneous extraction of facial features and age/gender prediction [32];
- Our EfficientNet-B0 and EfficientNet-B2[6] trained on the same VGGFace2 train-test split. Their fine-tuned versions are characterized by the state-of-the-art accu-

[2] https://github.com/deepinsight/insightface/tree/master/model_zoo

[3] https://github.com/cydonia999/VGGFace2-pytorch

[4] https://github.com/davidsandberg/facenet

[5] https://github.com/HSE-asavchenko/HSE_FaceRec_tf/tree/master/age_gender_identity

[6] https://github.com/HSE-asavchenko/face-emotion-recognition/tree/main/models/pretrained_faces

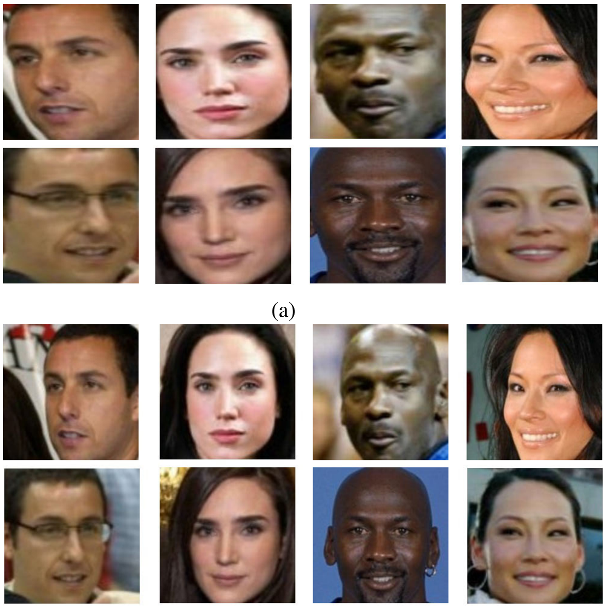


FIGURE 3. Sample images from the LFW dataset: (a) Cropped by RetinaFace without margins; (b) Cropped and aligned.

TABLE 1. The sizes of facial networks.

CNN	No. of weights, M.
PocketNetS-128 [44]	1.75
PocketNetM-256 [44]	1.01
MobileFaceNet [5]	1.22
InsightFace (IResNet-50) [1]	41.58
VGGFace2 (SENet-50) [4]	24.92
FaceNet (InceptionResNet) [2]	22.83
Multi-task MobileNet v1 [32]	3.33
EfficientNet-B0 [49]	3.89
EfficientNet-B2 [49]	16.46
Our SuperNet	8.75
Our Subnet 1, 865	4.52
Our Subnet 2, 865	3.16
Our Subnet 1, 765	4.67
Our Subnet 2, 765	3.00

racy for facial expression recognition on the AffectNet dataset [49].

In addition, we report the results of several well-known publicly available facial descriptors trained on the second version of preprocessing of the MS-Celeb-1M dataset, namely, MS1M-refine-v2:

- TensorFlow implementation[7] for MobileFaceNet [5];
- NAS-based official PocketNetS-128 and PocketNetM-256 [44] models "295672backbone" and "261556backbone".[8]

As the proposed method targets mobile applications, the memory size of the model is also essential. The network sizes (number of parameters) of the backbones of the above-mentioned neural networks (without the last fully-connected classification layer) are shown in Table 1.

VI. EXPERIMENTAL RESULTS

A. FACE VERIFICATION

The mean and standard deviation of the face verification metrics for cropped (Fig. 3a) and aligned (Fig. 3b) faces are shown in Table 2 and Table 3, respectively. The best results in each column are in bold, while the second and third-best metrics are underlined.

Here, first, modern facial descriptors (InsightFace, VGGFace2, FaceNet) are characterized by much better quality when compared to the mobile architectures (MobileNet/EfficientNet) trained by ourselves. However, our lightweight SuperNet and subnetworks obtained using the proposed framework (Fig. 1) show results competitive to the best-known models. It is remarkable because all these CNNs were trained on the same sets of cropped faces from the VGGFace2 dataset.

Second, VGGFace2 is not the best dataset for pre-training facial descriptors. All CNNs trained on other datasets are much better for facial alignment and loosely cropped faces (Table 3). However, they are not robust to small perturbations: the accuracy is significantly dropped (Table 2) for the facial regions at the output of the face detector (Fig. 3a). In the latter case, our models show the best metrics, even slightly better than for the aligned faces. The OFA SuperNet is more accurate than the very fast Subnet 2, but Subnet 1 offers practically the same accuracy as the SuperNet.

Such robustness is a critical property from the practical point of view [49]. Indeed, the background near the faces may vary drastically, so there is no guarantee that high quality is obtained for any background. Moreover, it may be impossible to rapidly get the facial key points and align faces in many mobile applications. Finally, it was demonstrated that such a robust network might be much better fine-tuned for other facial processing tasks, e.g., facial expression recognition [49].

B. FACE IDENTIFICATION

The mean and standard deviation of classification accuracy $\overline{A}$ are shown in Table 4. Though face verification and identification tasks are different, the results of this experiment are similar to the results of the previous investigation (Tables 2, 3). The InsightFace models are the most accurate for aligned faces but are much worse if the facial regions are cropped. It is especially noticeable for IResNet-50 architecture trained on the VGGFace2 dataset, characterized by one of the worst quality (82.34%) among all models. In the latter case, our models have at least a 3%-lower error rate than all existing facial descriptors. As the face identification task is more complex, Subnet 2, generated under strict time constraints, is 1.5-2% less accurate than our SuperNet.

C. ORIGINAL OFA VS. PROPOSED FRAMEWORK

This subsection compares the proposed framework with the original OFAMobileNetV3 SuperNet method. As mentioned

[7] https://github.com/sirius-ai/MobileFaceNet_TF
[8] https://github.com/fdbtrs/PocketNet/

TABLE 2. Verification results, cropped faces.

Dataset for pre-training	CNN	Accuracy, %	Val@1e-3, %	AUC	EER
MS1MV2	PocketNetS-128 [44]	91.75±0.82	51.10±3.12	0.97005	0.085
MS1MV2	PocketNetM-256 [44]	94.63±0.82	69.27±3.37	0.98511	0.055
MS1MV2	MobileFaceNet [5]	84.25±1.39	21.47±3.68	0.91043	0.162
VGGFace2	InsightFace (IResNet-50) [1]	96.28±0.85	82.90±2.19	0.99232	0.038
	VGGFace2 (SENet-50) [4]	99.08±0.37	95.87±1.78	0.99941	0.010
	FaceNet (InceptionResNet) [2]	98.97±0.57	96.87±1.49	0.99909	0.010
	Multi-task MobileNet v1 [32]	97.33±0.80	85.47±2.43	0.99599	0.027
	EfficientNet-B0 [49]	97.82±0.83	89.77±2.30	0.99685	0.023
	EfficientNet-B2 [49]	98.20±0.82	85.43±2.49	0.99819	0.017
	Our SuperNet	**99.35±0.44**	**98.47±1.04**	**0.99975**	**0.006**
	Our Subnet 1, 865	99.28±0.41	97.93±1.20	0.99965	0.007
	Our Subnet 2, 865	99.02±0.56	97.83±1.24	0.99957	0.010

TABLE 3. Verification results, aligned faces.

Dataset for pre-training	CNN	Accuracy, %	Val@1e-3, %	AUC	EER
MS1MV2	PocketNetS-128 [44]	99.40±0.41	99.03±0.85	0.99905	0.005
MS1MV2	PocketNetM-256 [44]	**99.57±0.30**	**99.27±0.71**	0.99935	**0.004**
MS1MV2	MobileFaceNet [5]	99.17±0.50	97.17±1.42	0.99932	0.008
VGGFace2	InsightFace (IResNet-50) [1]	99.22±0.37	98.60±0.98	0.99929	0.006
	VGGFace2 (SENet-50) [4]	99.37±0.37	97.87±1.26	0.99964	0.006
	FaceNet (InceptionResNet) [2]	99.28±0.53	96.33±1.86	0.99939	0.008
	Multi-task MobileNet v1 [32]	98.05±0.76	84.87±1.89	0.99742	0.021
	EfficientNet-B0 [49]	97.17±0.70	88.20±1.79	0.99619	0.028
	EfficientNet-B2 [49]	98.45±0.88	90.60±2.02	0.99873	0.015
	Our SuperNet	99.32±0.34	98.00±1.27	**0.99978**	0.006
	Our Subnet 1, 865	99.22±0.35	98.00±1.32	0.99970	0.008
	Our Subnet 2, 865	99.20±0.41	97.57±1.42	0.99955	0.010

TABLE 4. Face identification accuracy $\bar{A}$ (%).

Dataset for pre-training	CNN	Aligned faces	Cropped faces
MS1MV2	PocketNetS-128 [44]	99.63 ± 0.06	63.94 ± 3.47
MS1MV2	PocketNetM-256 [44]	**99.70 ± 0.07**	76.12 ± 3.03
MS1MV2	MobileFaceNet [5]	97.42 ± 0.76	44.23 ± 4.16
VGGFace2	InsightFace (IResNet-50) [1]	99.23 ± 0.16	82.34 ± 3.35
	VGGFace2 (SENet-50) [4]	97.21 ± 4.19	96.61 ± 2.02
	FaceNet (InceptionResNet) [2]	96.12 ± 3.41	96.57 ± 1.13
	Multi-task MobileNet v1 [32]	92.60 ± 4.01	89.37 ± 4.58
	EfficientNet-B0 [49]	94.07 ± 4.18	94.70 ± 4.67
	EfficientNet-B2 [49]	95.00 ± 3.81	91.53 ± 4.53
	Our SuperNet	98.97 ± 1.0	**99.12 ± 0.83**
	Our Subnet 1, 865	98.13 ± 2.55	98.71 ± 1.05
	Our Subnet 2, 865	96.89 ± 3.55	97.34 ± 2.18

in Subsection IV-A, the loss function has been modified to train the SuperNet to extract representative features. Hence, we compare subnetworks extracted from the OFA trained with the original loss function (softmax categorical cross-entropy) and the new one (categorical cross-entropy with label smoothing and ArcFace). Moreover, as the main innovation of this paper is the replacement of the MLP-based accuracy predictor with the architecture defined in the original OFA [8] to the NAC-based Algorithm 1, their comparison for our OFA is also presented here. Similarly to our LightGBM classifier, this MLP predictor was trained on the same dataset at the output of unit "2. Accuracy estimator" (Fig. 1).

The results for the subnetworks extracted with different time constraints and LUTs are shown in Table 5. The quality metrics are computed for face verification and identification tasks, but only the cropped faces are used. This study demonstrates the benefits of the proposed Algorithm 1 with the NAC compared to the MLP-based accuracy predictor. As one can notice, the proposed NAC increases the validation

TABLE 5. Comparison of the proposed approach with the original OFA on the LFW, cropped faces.

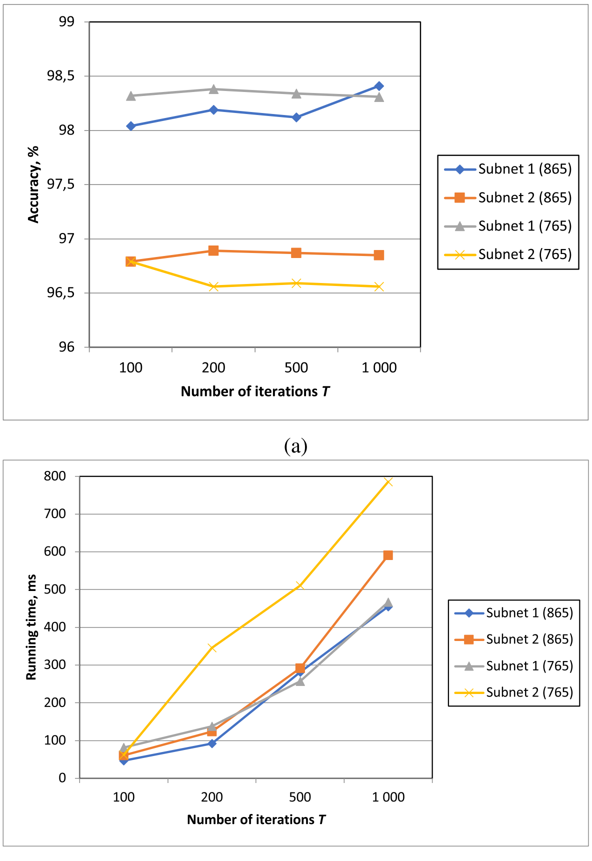
CNN				Verification		Identification
Device	Time constraint	OFA	Accuracy predictor	Accuracy, %	Val@1e-3, %	Accuracy, %
865	$t \leq 0.6 t_{ENet}$	Original [8]	MLP	98.45±0.51	96.34±1.64	96.58 ± 3.05
		Our	MLP	99.10±0.47	97.00±1.66	97.22 ± 3.11
		Our	Proposed (NAC)	99.22±0.35	98.00±1.32	98.13 ± 2.55
865	$t \leq 0.4 t_{ENet}$	Original [8]	MLP	98.39±0.54	96.04±1.80	96.05 ± 3.85
		Our	MLP	99.12±0.49	96.87±1.74	96.48 ± 4.13
		Our	Proposed (NAC)	99.20±0.41	97.57±1.42	96.89 ± 3.55
765	$t \leq 0.6 t_{ENet}$	Original [8]	MLP	98.51±0.43	96.17±1.64	96.67 ± 3.12
		Our	MLP	99.22±0.39	96.87±1.49	97.30 ± 2.86
		Our	Proposed (NAC)	99.30±0.40	98.03±1.33	98.34 ± 1.93
765	$t \leq 0.4 t_{ENet}$	Original [8]	MLP	98.25±0.69	96.96±1.82	96.39 ± 3.99
		Our	MLP	98.95±0.52	97.17±1.69	96.50 ± 4.08
		Our	Proposed (NAC)	99.07±0.46	96.97±1.70	96.64 ± 3.77

rate at FAR 0.001 and recognition accuracy up to 1.1% greater, especially if the time constraints are not too stringent. Otherwise, the number of potential subnetworks that satisfy the latency constraints is deficient. As a result, any search procedure will likely find models with approximately equal accuracy (compare the Subnets 2 generated for Snapdragon 765). It is necessary to highlight that the quality of the MLP predictor is relatively low, though it reached RMSE (root-mean-square error) 0.16% on the validation set. For example, it predicts the accuracy of Subnet 1 (865) on the testing part of the VGGFace2 to be equal to 99.72% while the absolute accuracy of this subnetwork is equal to 97.90%. A similar 1-2% prediction error is observed even in the original accuracy predictor from the OFA [8] on the ImageNet dataset. However, if we train the LightGBM predictor, it is much more precise: predicted accuracy on the VGGFace2 test set (98.47%) is approximately equal to the absolute accuracy (98.29%) of the extracted subnetwork.

D. ABLATION STUDY

In this Subsection, we conduct an ablation study to analyze the importance of the various hyperparameters in Algorithm 1. We estimate the face classification accuracy on the LFW dataset (aligned faces) and the time to run an evolutionary search on the server with Intel Core i9-10980XE (3.0GHz/24.75MB/18 cores) and 64Gb RAM. Four subnetworks have been generated for the LUTs from Snapdragon 865 and 765, and two latency constraints (Table 5). The dependencies of our quality measures on the number of iterations T, parents K_r, mutations M_r, population size P, and the mutation probability P_m are shown in Figs. 4-8. The following default values of hyperparameters were used: $T = 500$, $P = 100$, $K_r = 0.25$, $M_r = 0.5$, $P_m = 0.1$, so only one hyperparameter varied at a time to obtain each curve on these figures.

As expected, the running time of the evolutionary search linearly depends on the number of generations (Fig. 4b), but the classification accuracy does not have a similar trend.



(a)

(b)

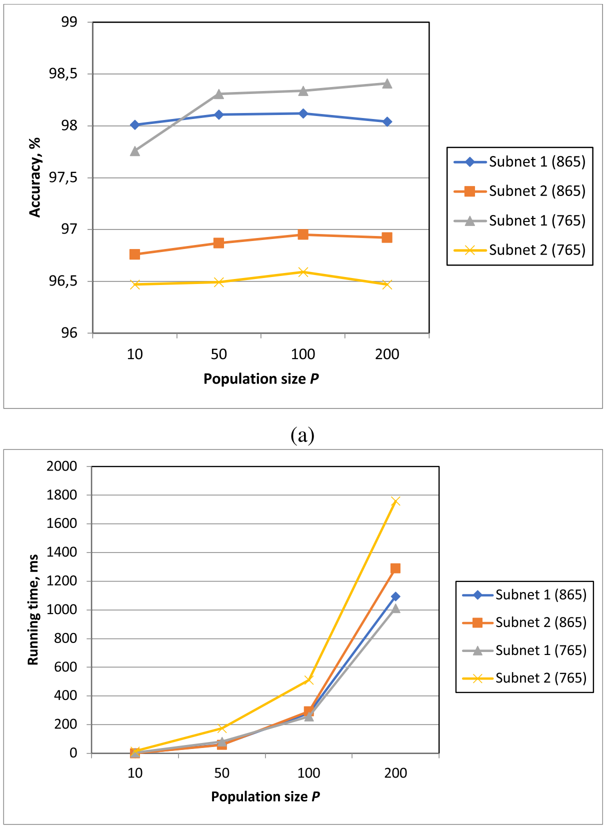
FIGURE 4. Dependence of (a) face identification accuracy on LFW, and (b) time of the evolutionary search for our subnetworks on the number of iterations *T*.

Hence, choosing a reasonable number of T of iterations is desirable to speed up the search for a subnetwork.

Secondly, the increase in population size leads to an even more significant increase in the time needed to find a suitable model (Fig. 5b). However, the accuracy is typically represented in a U-shaped curve (Fig. 5a). The optimal value of P is close to 100, at least for the chosen values of other hyperparameters.

TABLE 6. Inference time (ms) per one face.

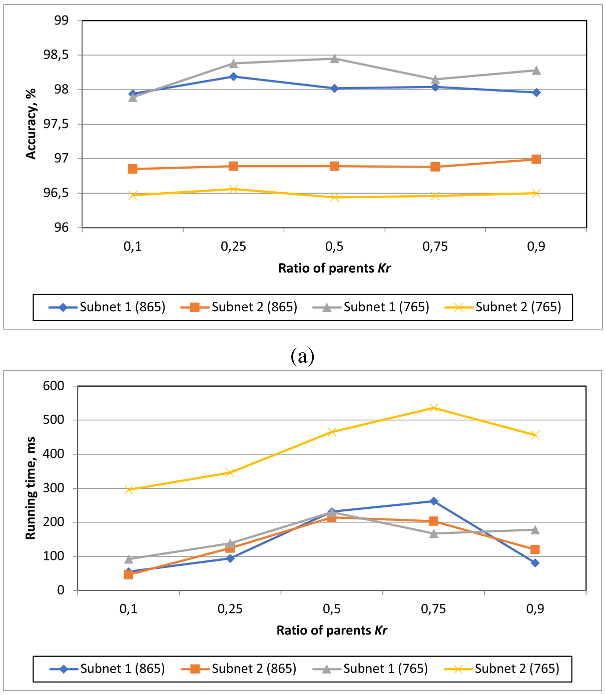
Device	CNN	PyTorch CPU	TFLite CPU	TFLite GPU
865	InsightFace (IResNet-50) [1]	-	203.75 ± 9.23	53.41 ± 0.71
	PocketNetM-128 [44]	284.00 ± 30.07	-	-
	PocketNetM-256 [44]	407.86 ± 48.37	-	-
	MobileFaceNet [5]	-	25.95 ± 1.51	4.75 ± 0.52
	Multi-task MobileNet v1 [32]	-	13.28 ± 0.63	4.78 ± 0.41
	Our Subnet 1	23.54 ± 1.50	11.89 ± 0.53	4.76 ± 0.45
	Our Subnet 2	19.78 ± 1.76	8.74 ± 0.58	3.55 ± 0.50
765	InsightFace (IResNet-50) [1]	-	507.12 ± 10.86	85.44 ± 2.26
	PocketNetM-128 [44]	715.63 ± 41.21	-	-
	PocketNetM-256 [44]	1036.11 ± 59.20	-	-
	MobileFaceNet [5]	-	39.08 ± 1.34	9.05 ± 0.47
	Multi-task MobileNet v1 [32]	-	33.07 ± 1.12	8.65 ± 1.05
	Our Subnet 1	64.74 ± 2.21	34.02 ± 1.36	9.15 ± 0.87
	Our Subnet 2	47.79 ± 2.09	22.82 ± 0.94	6.19 ± 0.72



(a)

(b)

FIGURE 5. Dependence of (a) face identification accuracy on LFW, and (b) time of the evolutionary search for our subnetworks on the population size *P*.



(a)

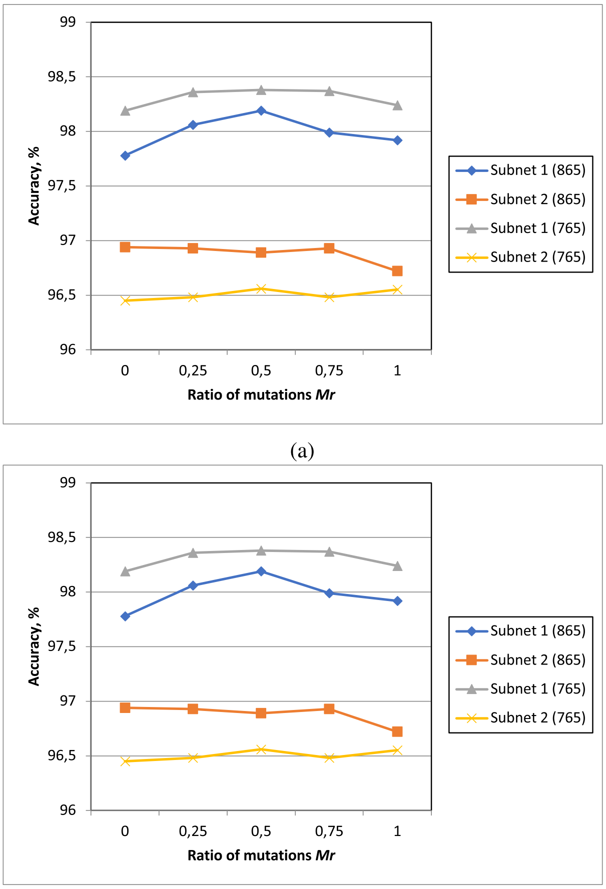
(b)

FIGURE 6. Dependence of (a) face identification accuracy on LFW, and (b) time of the evolutionary search for our subnetworks on the relative number of parents K_r.

Thirdly, the number of subnetworks that are used as parents should be chosen carefully. For example, its default value (25, i.e., $P \cdot K_r = 100 \cdot 0.25$) lets us reach the maximal accuracy in all cases except Subnet 1 (Snapdragon 865), for which it is better to have a more significant number of parents (Fig. 6a). Surprisingly, the running time is typically increased with the growth of K_r (Fig. 6b), but the time of search becomes much lower for a considerable number of parents ($100 \cdot 0.9 = 90$).

Fourthly, the default value of the number of subnetworks generated by a mutation in one generation ($P \cdot M_r = 100 \cdot 0.5 = 50$) works reasonably well. It does not reach the maximal accuracy only for one case (Subnet 2, Snapdragon 865), but the difference in the accuracy 0.04% is negligible (Fig. 7a). The running time also does not depend on M_r: the evolutionary search procedure becomes significantly faster only for the high number of mutations (Fig. 7b).

Finally, the probability of mutation in the depth of the network and kernel size and expand ratio of each block is



(a)

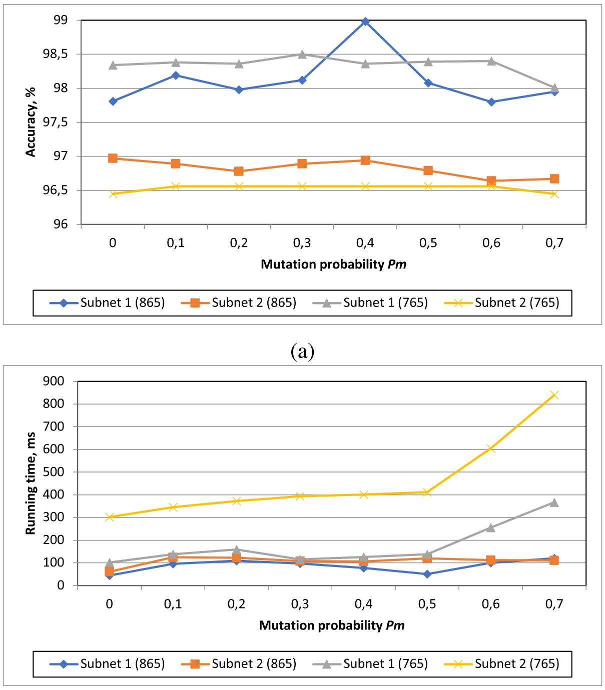
(b)

FIGURE 7. Dependence of (a) face identification accuracy on LFW, and (b) time of the evolutionary search for our subnetworks on the relative number of mutations M_r.

the most difficult hyperparameter to set. Its default value ($P_m = 0.1$) is not optimal in all four cases. It is important to emphasize that the value of P_m should not be very high, as the evolutionary search becomes too slow. Indeed, it is difficult to mutate most of the subnetwork and still satisfy the latency constraint.

E. RUNNING TIME ON MOBILE DEVICES

In this Subsection, our mobile demo application measured the inference time per one face (Fig. 2). Two Xiaomi mobile devices were utilized: Mi 10T Pro with Snapdragon 865 and Mi 10 Lite with Snapdragon 765g. Experimental results are summarized in Table 6. As one can notice, the InsightFace models are inappropriate for offline facial processing on a mobile device due to their very high accuracy. It is important to emphasize that though PocketNets models have a few parameters, they run very slowly in the PyTorch Mobile environment. These models seem to have some layers that should not be used on mobile devices. Even our slowest models ("Subnet 1") are as fast as MobileNet v1, while Subnet 2 is 1.5 times faster. Unit 6 in our pipeline (Fig. 1) is essential. Indeed, in contrast to TensorFlow Lite, PyTorch 1.9 does not support running on the GPU of a mobile device. Even the running time on the CPU of PyTorch models is twice as high as the equivalent TFLite model.



(a)

(b)

FIGURE 8. Dependence of (a) face identification accuracy on LFW, and (b) time of the evolutionary search for our subnetworks on the mutation probability P_m.

VII. CONCLUSION AND FUTURE WORKS

This paper proposes a novel engine (Fig. 1) to develop device-specific facial descriptors. The main **advantage** of our Algorithm 1 is the need for only several minutes to find a model given a latency constraint for a particular device. Another significant advantage is the high processing speed and accuracy. Indeed, it was experimentally demonstrated that extracted subnetworks process images faster than MobileNet v1 (Table 6) and reach near the state-of-the-art accuracy in both LFW's face verification (Table 2) and identification tasks (Table 4) using facial regions at the output of face detector without additional alignment and margins with potentially noisy background (Fig. 3a).

The subnetworks' generation pipeline in the original OFA [8] is based on an accuracy predictor, which focuses on solving the regression problem using MLP. Although the regression approach is relatively straightforward, it was demonstrated that shifting to GBDT-based binary classification can improve performance due to non-linear NAS space and the difficulty of precise prediction of the expected accuracy. As a result, using the proposed neural architecture comparator leads to up to 1% greater accuracy of generated models compared to the original accuracy predictor (Table 5).

The SuperNet is trained on the VGGFace2 dataset and made publicly available with several subnetworks and a demo Android application (Fig. 2). It may be used in any offline mobile services, such as clustering of family members and friends of a device's owner in a gallery of photos [26], [32]. The availability of TensorFlow Lite models makes it possible even to implement video-based face recognition on embedded or edge devices.

There are several **disadvantages** of the proposed approach. First, estimating the latency of a subnetwork running on a mobile GPU is impossible by summing the running times for each layer from the LUT (step 4 in Algorithm 1).

The second shortcoming of our Algorithm 1 is the absence of memory constraints in the generation of subnetworks, though it is known that the memory size of the network is also essential. As a result, several existing publicly-available facial models have much smaller sizes (Table 1), though the accuracy and latency of our subnetworks are significantly better.

One of the widely-used techniques to control the model's size is network compression [52], [53]. Moreover, it can influence the latency, as many embedded devices work much faster with int8-quantized models. From this point of view, our current models have a third limitation: they struggle with post-training quantization, so the accuracy degradation sometimes reaches 10-20%

Hence, in the future, it is necessary to incorporate the memory constraint into criteria (3), evolution search, and the architecture of SuperNet. For example, the penultimate layer of the OFAMobileNetV3 and all extracted subnetworks is the Conv2D layer that converts 1152-dimensional features to 1536-dimensional facial representations, which is relatively fast but has more than 1.5M parameters.

Moreover, it is necessary to apply more sophisticated quantization techniques and modify the proposed framework for a joint search of multi-AutoML with simultaneous quantization and pruning [43] and quantization-aware selection of candidates for efficiently compressed subnetworks.

Another research direction is the usage of more complex datasets to pre-train the SuperNet for our method. The difference in validation accuracies of the deepest (98.3%) and lightweight subnetworks (97.5%) is less than 1%, so choosing the most reliable facial descriptor that works in cross-domain settings is challenging.

ACKNOWLEDGMENT

The authors would like to thank Dmitry Maslov and Ildar Idrisov for their help with the implementation of training and model conversion software for the Once-for-All SuperNet.

REFERENCES

[1] J. Deng, J. Guo, T. Liu, M. Gong, and S. Zafeiriou, "Sub-center ArcFace: Boosting face recognition by large-scale noisy web faces," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2020, pp. 741–757.

[2] 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.

[3] A. V. Savchenko and N. S. Belova, "Statistical testing of segment homogeneity in classification of piecewise–regular objects," *Int. J. Appl. Math. Comput. Sci.*, vol. 25, no. 4, pp. 915–925, Dec. 2015.

[4] 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.

[5] S. Chen, Y. Liu, X. Gao, and Z. Han, "MobileFaceNets: Efficient CNNs for accurate real-time face verification on mobile devices," in *Proc. Chin. Conf. Biometric Recognit.* Cham, Switzerland: Springer, 2018, pp. 428–438.

[6] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "MnasNet: Platform-aware neural architecture search for mobile," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2815–2823.

[7] Q. Guo, X.-J. Wu, J. Kittler, and Z. Feng, "Differentiable neural architecture learning for efficient neural networks," *Pattern Recognit.*, vol. 126, Jun. 2022, Art. no. 108448.

[8] H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han, "Once-for-all: Train one network and specialize it for efficient deployment," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–15.

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

[10] W. Ali, W. Tian, S. U. Din, D. Iradukunda, and A. A. Khan, "Classical and modern face recognition approaches: A complete review," *Multimedia Tools Appl.*, vol. 80, no. 3, pp. 4825–4880, Jan. 2021.

[11] L. Best-Rowden, H. Han, C. Otto, B. F. Klare, and A. K. Jain, "Unconstrained face recognition: Identifying a person of interest from a media collection," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2144–2157, Dec. 2014.

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

[13] Q. Meng, S. Zhao, Z. Huang, and F. Zhou, "MagFace: A universal representation for face recognition and quality assessment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14220–14229.

[14] X. Yang, X. Jia, D. Gong, D.-M. Yan, Z. Li, and W. Liu, "LARNet: Lie algebra residual network for face recognition," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 11738–11750.

[15] A. V. Savchenko and L. V. Savchenko, "Towards the creation of reliable voice control system based on a fuzzy approach," *Pattern Recognit. Lett.*, vol. 65, pp. 145–151, Nov. 2015.

[16] E. Z. Xu, Z. Song, S. Tsutsui, C. Feng, M. Ye, and M. Z. Shou, "AVA-AVD: Audio-visual speaker diarization in the wild," in *Proc. 30th ACM Int. Conf. Multimedia (MM)*, Oct. 2022, pp. 3838–3847.

[17] A. V. Savchenko, "Phonetic words decoding software in the problem of Russian speech recognition," *Autom. Remote Control*, vol. 74, no. 7, pp. 1225–1232, Jul. 2013.

[18] D. Maslov and I. Makarov, "Online supervised attention-based recurrent depth estimation from monocular video," *PeerJ Comput. Sci.*, vol. 6, p. e317, Nov. 2020.

[19] A. V. Savchenko and Y. I. Khokhlova, "About neural-network algorithms application in viseme classification problem with face video in audiovisual speech recognition systems," *Opt. Memory Neural Netw.*, vol. 23, no. 1, pp. 34–42, Jan. 2014.

[20] I. Makarov, M. Bakhanova, S. Nikolenko, and O. Gerasimova, "Self-supervised recurrent depth estimation with attention mechanisms," *PeerJ Comput. Sci.*, vol. 8, p. e865, Jan. 2022.

[21] W. Huang, J. Gu, and Y. Guo, "Depth-aware object tracking with a conditional variational autoencoder," *IEEE Access*, vol. 9, pp. 94537–94547, 2021.

[22] I. Makarov, V. Aliev, and O. Gerasimova, "Semi-dense depth interpolation using deep convolutional neural networks," in *Proc. 25th ACM Int. Conf. Multimedia (MM)*. New York, NY, USA: FX Palo Alto Laboratory, Oct. 2017, pp. 1407–1415.

[23] Z. Tasneem, G. Milione, Y.-H. Tsai, X. Yu, A. Veeraraghavan, M. Chandraker, and F. Pittaluga, "Learning phase mask for privacy-preserving passive depth estimation," in *Proc. 17th Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2022, pp. 504–521.

[24] I. Makarov, V. Aliev, O. Gerasimova, and P. Polyakov, "Depth map interpolation using perceptual loss," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*. New York, NY, USA: Ecole Centrale de Nantes, Oct. 2017, pp. 93–94.

[25] A. Korinevskaya and I. Makarov, "Fast depth map super-resolution using deep neural network," in *Proc. 17th IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*. New York, NY, USA: TU Munich, Oct. 2018, pp. 117–122.

[26] A. V. Savchenko, K. V. Demochkin, and I. S. Grechikhin, "Preference prediction based on a photo gallery analysis with scene recognition and object detection," *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108248.

[27] A. D. Sokolova, A. S. Kharchevnikova, and A. V. Savchenko, "Organizing multimedia data in video surveillance systems based on face verification with convolutional neural networks," in *Proc. Int. Conf. Anal. Images, Social Netw. Texts (AIST)*. Cham, Switzerland: Springer, 2018, pp. 223–230.

[28] S. Gong, X. Liu, and A. K. Jain, "Mitigating face recognition bias via group adaptive classifier," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3413–3423.

[29] Y. Li, Z. Yang, Y. Chen, D. Yang, R. Liu, and L. Jiao, "Occluded person re-identification method based on multiscale features and human feature reconstruction," *IEEE Access*, vol. 10, pp. 98584–98592, 2022.

[30] X. Gao, Y. Sun, Y. Xiao, Y. Gu, S. Chai, and B. Chen, "Adaptive random down-sampling data augmentation and area attention pooling for low resolution face recognition," *Expert Syst. Appl.*, vol. 209, Dec. 2022, Art. no. 118275.

[31] P. C. Neto, J. R. Pinto, F. Boutros, N. Damer, A. F. Sequeira, and J. S. Cardoso, "Beyond masks: On the generalization of masked face recognition models to occluded face recognition," *IEEE Access*, vol. 10, pp. 86222–86233, 2022.

[32] A. V. Savchenko, "Efficient facial representations for age, gender and identity recognition in organizing photo albums using multi-output ConvNet," *PeerJ Comput. Sci.*, vol. 5, p. e197, Jun. 2019.

[33] M. Rouhsedaghat, Y. Wang, S. Hu, S. You, and C.-C.-J. Kuo, "Low-resolution face recognition in resource-constrained environments," *Pattern Recognit. Lett.*, vol. 149, pp. 193–199, Sep. 2021.

[34] 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.

[35] T.-J. Yang, A. Howard, B. Chen, X. Zhang, A. Go, M. Sandler, V. Sze, and H. Adam, "NetAdapt: Platform-aware neural network adaptation for mobile applications," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 285–300.

[36] A. Howard, M. Sandler, B. Chen, W. Wang, L. 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.

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

[38] C. Xue, M. Hu, X. Huang, and C.-G. Li, "Automated search space and search strategy selection for AutoML," *Pattern Recognit.*, vol. 124, Apr. 2022, Art. no. 108474.

[39] G. Bender, P.-J. Kindermans, B. Zoph, V. Vasudevan, and Q. Le, "Understanding and simplifying one-shot architecture search," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 550–559.

[40] Z. Guo, X. Zhang, H. Mu, W. Heng, Z. Liu, Y. Wei, and J. Sun, "Single path one-shot neural architecture search with uniform sampling," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2020, pp. 544–560.

[41] Y. Chen, Y. Guo, Q. Chen, M. Li, W. Zeng, Y. Wang, and M. Tan, "Contrastive neural architecture search with neural architecture comparators," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9497–9506.

[42] Y. Hu, X. Wang, L. Li, and Q. Gu, "Improving one-shot NAS with shrinking-and-expanding supernet," *Pattern Recognit.*, vol. 118, Oct. 2021, Art. no. 108025.

[43] T. Wang, K. Wang, H. Cai, J. Lin, Z. Liu, H. Wang, Y. Lin, and S. Han, "APQ: Joint search for network architecture, pruning and quantization policy," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2075–2084.

[44] 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.

[45] X. Wang, S. Wang, C. Chi, S. Zhang, and T. Mei, "Loss function search for face recognition," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 10029–10038.

[46] H. Gu, J. Li, G. Fu, M. Yue, and J. Zhu, "Loss function search for person re-identification," *Pattern Recognit.*, vol. 124, Apr. 2022, Art. no. 108432.

[47] A. V. Savchenko, "Fast inference in convolutional neural networks based on sequential three-way decisions," *Inf. Sci.*, vol. 560, pp. 370–385, Jun. 2021.

[48] Y. Zhu, Y. Liang, K. Tang, and K. Ouchi, "SC-NET: Spatial and channel attention mechanism for enhancement in face recognition," in *Proc. 5th Int. Conf. Inf. Comput. Technol. (ICICT)*, Mar. 2022, pp. 166–172.

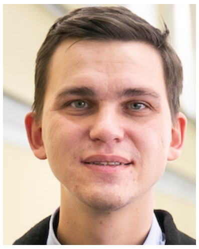
[49] A. V. Savchenko, L. V. Savchenko, and I. Makarov, "Classifying emotions and engagement in online learning based on a single facial expression recognition neural network," *IEEE Trans. Affect. Comput.*, vol. 13, no. 4, pp. 2132–2143, Oct. 2022.

[50] A. Rajpal, K. Sehra, R. Bagri, and P. Sikka, "XAI-FR: Explainable AI-based face recognition using deep neural networks," *Wireless Pers. Commun.*, vol. 129, no. 1, pp. 663–680, Mar. 2023.

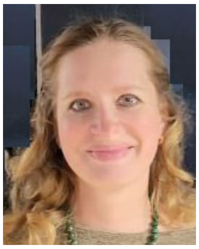
[51] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-shot multi-level face localisation in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5202–5211.

[52] H. Bai, M. Cao, P. Huang, and J. Shan, "BatchQuant: Quantized-for-all architecture search with robust quantizer," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 1074–1085.

[53] A. M. Grachev, D. I. Ignatov, and A. V. Savchenko, "Neural networks compression for language modeling," in *Proc. 7th Int. Conf. Pattern Recognit. Mach. Intell. (PReMI)*. Cham, Switzerland: Springer, 2017, pp. 351–357.



ANDREY V. SAVCHENKO received the B.S. degree in applied mathematics and informatics from Nizhny Novgorod State Technical University, Nizhny Novgorod, Russia, in 2006, the Ph.D. degree in mathematical modeling and computer science from the State University Higher School of Economics, Moscow, Russia, in 2010, and the Dr. of Science degree in system analysis and information processing from Nizhny Novgorod State Technical University, in 2016. Since 2022, he has been with the Sber AI Laboratory, where he is currently the Scientific Director. He is also a Leading Research Fellow with the Laboratory of Algorithms and Technologies for Network Analysis, HSE University, Nizhny Novgorod. He has authored or coauthored one monograph and more than 50 articles. His research interests include statistical pattern recognition, image classification, and biometrics.



LYUDMILA V. SAVCHENKO received the Specialist degree in applied mathematics and informatics from Nizhny Novgorod State Technical University, Nizhny Novgorod, Russia, in 2008, and the Ph.D. degree in system analysis and information processing from Voronezh State Technical University, in 2017. Since 2018, she has been with HSE University, Nizhny Novgorod, where she is currently an Associate Professor with the Department of Information Systems and Technologies. She is also a Senior Research Fellow with the Laboratory of Algorithms and Technologies for Network Analysis, HSE University. Her current research interests include speech processing and e-learning systems.

ILYA MAKAROV received the Specialist degree in mathematics from Lomonosov Moscow State University, Moscow, Russia, and the Ph.D. degree in computer science from the University of Ljubljana, Ljubljana, Slovenia.

Since 2011, he has been a Lecturer with the School of Data Analysis and Artificial Intelligence, HSE University, where he was the School Deputy Head, from 2012 to 2016. He is currently an associate professor and a senior research fellow. He was also the Program Director of BigData Academy MADE from VK and a Researcher with the Samsung-PDMI Joint AI Center, St. Petersburg Department, V. A. Steklov Mathematical Institute, Russian Academy of Sciences, Saint Petersburg, Russia. He is also a Senior Research Fellow with the Artificial Intelligence Research Institute (AIRI), Moscow, where he leads the research in industrial AI. He became the Head of the AI Research Center and the Data Science Tech Master Program in NLP, National University of Science and Technology MISIS.