IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 14, 2020, accepted September 9, 2020, date of publication September 14, 2020, date of current version September 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3024116

Deep Convolutional Neural Networks for Unconstrained Ear Recognition

HAMMAM ALSHAZLY[1,2], CHRISTOPH LINSE[1], ERHARDT BARTH[1], (Member, IEEE), AND THOMAS MARTINETZ[1], (Senior Member, IEEE)

[1]Institute for Neuro- and Bioinformatics, University of Lübeck, 23562 Lübeck, Germany
[2]Department of Mathematics, Faculty of Science, South Valley University, Qena 83523, Egypt

Corresponding author: Hammam Alshazly (alshazly@inb.uni-luebeck.de)

The work of Hammam Alshazly was supported by the Bundesministerium für Bildung und Forschung (BMBF) through the KI-Lab Project. The work of Christoph Linse was supported by the Bundesministeriums für Wirtschaft und Energie (BMWi) through the Mittelstand 4.0-Kompetenzzentrum Kiel Project.

ABSTRACT This paper employs state-of-the-art Deep Convolutional Neural Networks (CNNs), namely AlexNet, VGGNet, Inception, ResNet and ResNeXt in a first experimental study of ear recognition on the unconstrained EarVN1.0 dataset. As the dataset size is still insufficient to train deep CNNs from scratch, we utilize transfer learning and propose different domain adaptation strategies. The experiments show that our networks, which are fine-tuned using custom-sized inputs determined specifically for each CNN architecture, obtain state-of-the-art recognition performance where a single ResNeXt101 model achieves a rank-1 recognition accuracy of 93.45%. Moreover, we achieve the best rank-1 recognition accuracy of 95.85% using an ensemble of fine-tuned ResNeXt101 models. In order to explain the performance differences between models and make our results more interpretable, we employ the t-SNE algorithm to explore and visualize the learned features. Feature visualizations show well-separated clusters representing ear images of the different subjects. This indicates that discriminative and ear-specific features are learned when applying our proposed learning strategies.

INDEX TERMS Ear recognition, biometrics, deep learning, convolutional neural networks, transfer learning, feature visualization.

I. INTRODUCTION

Ear recognition refers to the process of automated human recognition based on the physical characteristics of the ears. It is highly relevant to a broad range of application domains such as forensics, surveillance, identity checks and unlocking user's devices. Based on the unique structure of the ear shape, ear images can provide a rich source of biometric information for constructing successful recognition systems. In addition, there are several desirable characteristics of the human ears which include: ease of capture from a distance, stability over time, ability to identify identical twins [1], and being insensitive to emotions and facial expressions [2], [3]. Given these appealing features we can build and develop reliable recognition systems on numerous devices in a non-intrusive and non-distracting manner [4]–[6]. Nevertheless, an accurate recognition can be a challenging task when ear images are acquired in unconstrained environments where various appearance variations and illumination changes need to be considered [7].

The early work of ear recognition research has demonstrated significantly improved performance especially for ear images collected under controlled conditions [8]. Most of these techniques employed manual feature engineering (a.k.a. handcrafted) methods to describe the important features of ear images. The obtained features were then used to train a traditional classifier to learn the specific patterns in the extracted features to discriminate individuals. The performance of these ear recognition techniques is greatly affected by the robustness of the feature extraction method and the effectiveness of the employed classifier. In essence, these techniques suffer from two key limitations. On one hand, extracting the relevant features manually from images requires individuals with a strong knowledge of the specific domain, and it is a time-consuming process. On the other hand, the performance of these methods drops with the increased level of appearance variability in the given images. Therefore, failing to address these limitations results in

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

performance deterioration, especially when recognizing ear images under uncontrolled imaging conditions.

In recent years, deep learning algorithms and more specifically deep Convolutional Neural Networks (CNNs) have lead to breakthroughs in many application domains including image classification [9]–[13], object detection [14]–[17] and biometric recognition [18]. These improvements are the result of several factors including the availability of tremendous amounts of labeled data, powerful hardware (i.e. GPUs) for accelerating computations, well-designed deep network architectures, effective optimization techniques and the technical improvement in training deep networks. Besides being scalable supervised learning techniques, deep CNNs perform the feature extraction and classification by training the entire system in an end-to-end manner and obviate the manual feature extraction. However, training deep CNNs requires optimizing a large number of trainable parameters (millions) and large-scale labeled datasets. In addition, collecting such amounts of data may be expensive for some real-world applications and as a consequence limiting the high potentials of deep models.

An effective approach to address the above-mentioned limitations is to utilize transfer learning [19], [20]. It is a strategy in which the knowledge learned by a deep CNN on a given task and dataset is transferred or utilized to initialize deep CNNs to tackle different but related tasks and new datasets. Nowadays, transfer learning has become the most viable solution for addressing the challenging visual recognition tasks. In this work, we address the problem of recognizing ear images collected under unconstrained conditions through utilizing transfer learning with deep CNNs, and report the results of the first ear recognition experiments on the EarVN1.0 dataset. The research efforts and outcomes have resulted in several important contributions that are summarized below:

- We present the first experimental study of ear recognition on the unconstrained EarVN1.0 dataset. To this end, we employ state-of-the-art deep CNN architectures of different depth and provide comparative evaluation and analysis of their recognition performance.
- Two seminal deep CNN architectures (InceptionV3 and ResNeXt) are evaluated for the first time in ear recognition experiments. Extensive experiments and a profound analysis of their performance and computational complexity are presented. We also leverage the recently proposed layer-wise adaptive large batch optimization technique called LAMB [21] to train all networks. The LAMB optimizer has demonstrated effectiveness in training deep networks outperforming adaptive optimization techniques such as Adam optimizer [22] as reported in [21], [23].
- We propose a two-step fine-tuning strategy for CNN architectures with more than one fully connected layer. We experiment with training the networks with fixed input size and custom input size determined for each network to preserve the aspect ratio of ear images. Our networks fine-tuned with custom size inputs obtain state-of-the-art results, with a rank-1 accuracy of 93.45%, indicating the effectiveness of our proposed strategy.
- We explore the effectiveness of deep ensembles of independently fine-tuned CNNs to improve the overall recognition accuracy. A relative improvement in accuracy above 2% is attained for each of the considered networks compared to using a single network.
- We provide visualizations of the learned features by the deep models under each learning strategy. The visualizations provide a clear evidence that our models have learned more discriminative features. This makes the obtained results more interpretable.

The next section reviews the related work. Section III describes the considered deep CNNs. Section IV explains the different transfer learning strategies. The experimental setup, dataset and evaluation metrics are mentioned in Section V. A comparative analysis of the obtained results and visualizations of the extracted features are reported in Section VI. Finally, Section VII draws the paper's conclusion.

II. RELATED WORK

The ear recognition field has witnessed an increasing interest during the last few years and nearly perfect recognition rates have been attained under constrained conditions [24]–[27]. The proposed techniques were developed and evaluated using small ear datasets gathered under laboratory-like settings with limited variations in lighting, head poses and occlusions. Even though these approaches and datasets contributed to promoting the field, the performance of these techniques showed deterioration when considering real-life scenarios in which the ear images are significantly affected by variations in head poses, illumination, blurring, occlusions and other factors [8], [28], [29].

The research interest to conduct ear recognition experiments under unconstrained settings is motivated by two main aspects. First, the availability of public ear image datasets that were collected under real-world settings. Second, the rapid and continuous improvements in representation learning methods and more specifically deep CNN architectures. These CNNs can automatically learn more discriminative features from the given images without any human supervision, and achieve state-of-the-art performance on various vision tasks.

A number of ear image datasets collected under unconstrained conditions has been released by the research community. These datasets have different characteristics based on their sources, appearance variations and the number of subjects. One of the early datasets that introduced a wide range of image variability is the West Pomeranian University of Technology (WPUT) ear dataset [30]. The dataset consisted of 2,071 ear images for 501 subjects of various age. The ear images provided a significant range of appearance variability in illumination, head poses and occlusions.

The Annotated Web Eear (AWE) dataset was publicly released in [8] and consisted of 1,000 ear images for 100 subjects. The images were collected from the Internet for a list of known celebrities and were tightly cropped around the ear. An extension of AWE dataset (AWEx) was presented in [31] with a total of 4,104 images for 346 subjects. The Unconstrained Ear Recognition Challenge (UERC) dataset [32] dataset was gathered as a further extension to the AWE and AWEx datasets for a specific ear recognition competition. The dataset contained 11,804 ear images and was divided into two main splits, 2,304 images belonging to 166 subjects for training and 9,500 images belonging to 3540 subjects for testing.

The in-the-wild ear dataset was introduced in [33] with 2,058 ear images for 231 subjects. The ear images were cropped from three public datasets built specifically for face recognition in the wild. A similar dataset is the Webears dataset [34], which was collected from the Internet and contained 1,000 ear images acquired by various devices and under different lighting conditions, occlusions, head poses and varying image resolutions. Zhang *et al.* introduced the USTB-Helloear [35] dataset. The ear images were extracted from a video sequence. The images showed different pose variations and levels of ear occlusion to reflect the uncontrolled conditions.

The recently released EarVN1.0 [36] dataset contained 28,412 ear images collected from the Internet in a similar way as the abovementioned datasets. It is considered one of the largest datasets of ear images collected under unconstrained conditions. Unlike all the above described datasets, to our knowledge, there are no ear recognition experiments carried out on this dataset.

An earlier work reported on CNNs for ear recognition was introduced in [37], [38]. The authors used small datasets with limited appearance variations to conduct their experiments. The obtained results indicated superior performance of the CNN-based models over traditional methods with respect to various image attributes.

In [39], the authors demonstrated the potential to train deep models with limited amount of ear images. Extensive image augmentation techniques were applied to obtain additional training samples from existing ones. The artificially introduced appearance variations helped to improve the generalization abilities of the obtained models. The AWE dataset was considered for evaluating the performance of the deep models and to report their recognition performance.

A recent study was conducted by Khadili and Benzaoui [40] for recognizing ear images collected under constrained and unconstrained conditions. A new framework was proposed to address the problem of using gray-scale ear images in the test phase for deep models trained with colored images. The authors employed conditional deep Generative Adversarial Networks (GANs) for colorizing the gray-scale ear images, and CNN models for the recognition task. The reported results indicated a significant impact of color information to achieve better recognition performance.

In [31], the authors investigated and analyzed different aspects of the recognition techniques and the impact of various factors on their performance. The analysis covered ear recognition models constructed using both engineered and CNN-based feature extraction methods. The CNN-based models are fine-tuned with additional ear images and then applied to extract useful features from the benchmark dataset. The extracted feature vectors are then classified based on the cosine similarity metric. The reported results indicated superior performances for recognition models utilizing CNN features, see also [41], [42].

Ensemble learning, which combines the predictions of different deep models, have also been employed in several studies to boost recognition accuracy under unconstrained settings. In [43], the authors proposed averaging the prediction of several fine-tuned deep networks. They used the AWE dataset to train and test models. They concluded that constructing an ensemble of fine-tuned networks increases the recognition accuracy independent of the dataset size. Zhang *et al.* [44] fine-tuned pretrained deep CNNs using ear images of different scales to obtain a multi-scale ear image representation. Then, they assembled three fine-tuned CNN models using different scales of ear images. They utilized the USTB-Helloear [35] for fine-tuning models and the AWE dataset for testing. Alshazly *et al.* [45] proposed ensembles of fine-tuned deep networks of various depths to improve the recognition performance of single models. The reported results using constrained and unconstrained ear images indicated a relative improvement above 4% when using deep ensembles.

The first Unconstrained Ear Recognition Challenge (UERC) [32] was organized in 2017 to evaluate the advancement in ear recognition technology using a unified experimental protocol on the UERC ear dataset. The participants and organizers submitted eight ear recognition models for evaluation. A comparative analysis was conducted for all the submitted techniques along with their sensitivity to different image attributes. The reported results indicated the sensitivity of all approaches to specific head rotations and their performance deteriorated with scale.

A second round of the UERC competition was held in 2019 [46]. The evaluation was performed using the experimental protocol and dataset partitions for training and testing as in [32]. An extensive analysis was conducted to evaluate different aspects of ear recognition models including their sensitivity to ear occlusions, variability in image resolution and performance bias towards a specific gender. Although the submitted approaches achieved competitive performance and indicated improvement over the 2017 models, they showed sensitivity to ear occlusions and inadequate image resolution.

Our study complements the body of existing work on unconstrained ear recognition and presents the results of the first conducted experiments on the EarVN1.0 dataset [36]. We also evaluate for the first time new deep CNN models (InceptionV3 and ResNeXt) for ear recognition. Moreover, we provide a comparative evaluation and analysis for

numerous top-performing deep CNN architectures with various learning strategies. Furthermore, we visualize the learned features for each learning strategy to examine the models' visual knowledge and make our results more interpretable.

III. DEEP NETWORKS

Deep CNNs have recently evolved as the cornerstone algorithm for a wide variety of computer vision tasks. Their architectural design has been developed in various ways, from repeating stacks of convolutional blocks to highly modularized network architectures, in order to improve their representational capabilities and reduce their computational complexity. This section describes the deep CNN architectures considered for our study. In order to cover a wide spectrum of network designs we have chosen AlexNet [9], VGGNet [10], InceptionV3 [12], ResNet [11] and ResNeXt [47] for our experiments. These networks have been proposed to classify the ImageNet dataset [48], which has 1000 different classes. We introduce some changes and adjust the last layer(s) of each network to suit the number of subjects in the EarVN1.0 dataset and to accept images with arbitrary sizes for preserving their aspect ratios. A brief description of each architecture and the introduced changes are mentioned in the following subsections.

A. AlexNet

AlexNet [9] is considered a deep CNN architecture compared with previous CNNs such as LeNet-5 [49], and is the winner of the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC-2012) for image classification [50]. As a result, AlexNet has been applied to numerous recognition tasks including ear recognition [39], [42], [51], [52].

The network consists of eight weight layers: five convolution and three fully connected layers. The first, second, and fifth convolutional layers are followed by overlapping max-pooling operations of size 3×3 and a stride of 2 to reduce the width and height of the output volume. The first convolutional layer accepts RGB input images of size $227 \times 227 \times 3$ and applies 96 kernels of size $11 \times 11 \times 3$. The second convolutional layer filters the pooled outputs of the first layer with 256 kernels of size $5 \times 5 \times 96$. The third convolutional layer applies 384 kernels of size $3 \times 3 \times 256$ to the pooled outputs of the second layer. The fourth convolutional layer applies 384 kernels of size $3 \times 3 \times 384$, and the last convolutional layer applies 256 kernels of size $3 \times 3 \times 384$. The first two fully connected layers have 4096 neurons each, whereas the last fully connected layer has 1000 neurons matching the 1000 classes of ImageNet dataset.

The Rectified Linear Unit (ReLU) nonlinearity [53] is applied after each convolutional and fully connected layer, which makes training deep CNNs much faster. The responses of the first and second convolutional layers are locally normalized before the pooling operations. To reduce overfitting the authors employed two effective regularization methods: data augmentation and dropout [54].

For our experiments we use a variant of the AlexNet architecture proposed in [55]. We reduce the number of convolutional filters in the first, second and forth convolutional layers to 64, 192 and 256 instead of 96, 256 and 384, respectively. The last pooling layer is replaced by an adaptive average pooling layer, which makes the network suitable for arbitrary-sized input images. We found that 201×297 is an optimal input size for ear images from the considered dataset, which allows the filters to be tiled appropriately. Additionally, the number of neurons in the first two fully connected layers are reduced to half, and the last fully connected layer is replaced by a new one, which has 164 neurons matching the number of subjects in the EarVN1.0 dataset. Batch normalization layers are also applied after each ReLU activation. In order to combat overfitting we apply data augmentation techniques and dropout with 50% chance in the first two fully connected layers.

B. VGGNet

The Visual Geometry Group networks (VGGNets) [10] represent a class of very deep CNNs and top-performers in the ILSVRC-2014 for image recognition and object localization [50]. The authors investigated the network's depth on the recognition accuracy using a network depth from 11 to 19 layers. VGGNets have been applied to improve recognition performance on challenging image datasets including the unconstrained ear image datasets [32], [43], [44], [51].

VGGNets have brought two important characteristics that distinguish them from previous CNNs such as AlexNet [9]. First, small 3×3 receptive fields are used throughout the entire network. By stacking multiple convolutional layers, larger receptive fields such as 5×5 or 7×7 can be covered while the number of trainable parameters is significantly reduced. Second, multiple layers of identical characteristics are stacked to build deeper networks.

The VGGNet architectures consist of five convolutional blocks and three fully connected layers. Each convolutional block is followed by a max-pooling operation. To substitute the reduction in spatial dimension, the number of filters is doubled in the next convolutional block (going from 64 filters in the first block to 512 in the fourth and fifth blocks). The last max-pooling layer is followed by three fully connected layers with 4096, 4096, and 1000 neurons. More details are given in [10].

For our study we consider the 16- and 19-layer VGGNet architectures. Hereinafter, we use VGG16 and VGG19 to indicate the two architectures. The main difference between VGG16 and VGG19 is the additional convolutional layer in the third, fourth, and fifth blocks. The fully connected layers are identical in both architectures. For our experiments we introduce some additional modifications. First, replacing the last max-pooling layer with an adaptive average pooling layer before the first fully connected layer, which makes the networks applicable to arbitrary input size. Consequently, we found that an input size of 192×288 achieves the best results on the EarVN1.0 dataset. Second, we reduce

the size of the first two fully connected layers to half (e.g., 2048 instead of 4096) to combat overfitting. Third, we replace the last fully connected layer with a new one to match the 164 classes of the EarVN1.0 dataset.

C. INCEPTION

The Inception family of networks is a class of very deep CNN architectures developed by engineers and researchers at Google [56]. The first network architecture was introduced as GoogLeNet (a.k.a Inception-V1) in [56], and won the ILSVRC-2014 challenges for image classification and detection [50]. The architecture has been modified in various ways such as introducing batch normalization in [57] (Inception-V2), and factorizing convolutions of large filter sizes in [12] (Inception-V3).

The Inception architecture is quite different from the sequential CNN architectures, such as AlexNet and VGGNet in which each layer accepts only one input and produces only one output. In contrary, the Inception network consists of small building blocks called Inception modules (see Figure 1) that are stacked on top of each other along with conventional convolution and max-pooling layers to form the overall architecture. The idea was inspired by the network-in-network approach proposed in [58].

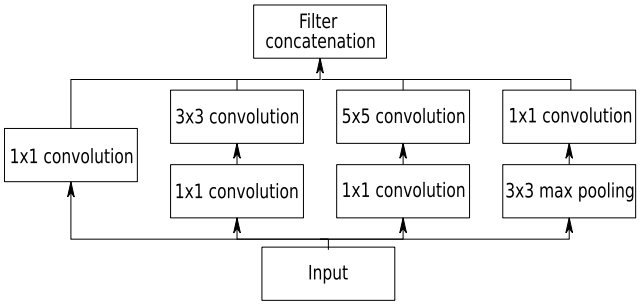

FIGURE 1. The Inception module as first introduced in [56].

The Inception module illustrated in Figure 1 was first introduced in [56]. It accepts an input from a previous layer, and then branches into four different paths each performing a specific operation. The input goes through 1×1, 3×3 and 5×5 convolutions as well as a max-pooling operation. Then, the different outputs are concatenated along the channel dimension as an output. There is also 1×1 convolution before the expensive 3×3 and 5×5 convolutions to reduce the dimensionality and avoid the computational bottlenecks and allows the network to go deeper. Essentially, by learning all 1×1, 3×3, and 5×5 convolutions we can consider an Inception module as a multi-level feature extractor.

In our study we consider the InceptionV3 model. The model consists of 42 weight layers and accepts RGB images of 299×299. However, in our experiments we found that an input size of 171×235 achieves the best results. While variants of the Inception network (InceptionV1) have been used in various recognition tasks including ear recognition [44], [51], to the best of our knowledge, InceptionV3 has not been utilized in any ear recognition experiments. Due to its superior performance as the first runner-up of the ILSVRC-2015 competition [50], it has been selected for our experiments.

D. ResNet

Residual networks (ResNets) [11] represent a class of extremely deep CNN architectures, which won the three ILSVRC-2015 competitions for object recognition, detection and localization [50]. The network depth has been empirically argued to be a crucial factor to improve the network's representational power. However, with the increased depth two major issues arise: the vanishing/exploding gradients and performance degradation [11]. ResNets addressed the problems by using skip connections that prevent information loss as the network goes deeper.

The ResNet architecture shares design similarity with the VGGNet. First, by stacking multiple building blocks of the same topology to construct very deep networks. Second, by using small 3×3 kernels in the convolutional layers to reduce the number of learnable parameters which require optimization during training. However, it is different from VGGNet in using the skip connection as identity mappings. Moreover, the ResNet architecture is considered a fully convolutional network as the convolution operations are employed to not only learn discriminative filters, but to reduce the spatial dimensions instead of the pooling layers. Throughout the entire ResNet architecture only two pooling layers are used. The first is a max-pooling layer after the first convolutional layer, and the second is an average-pooling layer at the end of the network before the softmax.

The cornerstone in the ResNet architecture is the residual module, which is depicted in Figure 2. The module accepts an input and then branches into two paths. The left path performs a series of 1×1 and 3×3 convolution, batch normalization and ReLU activation. The right path is an identity mapping which connects the module's input through an addition operation with the output of the left path. A deeper network can be constructed by stacking multiple ResNet blocks along with other conventional convolution and pooling layers.

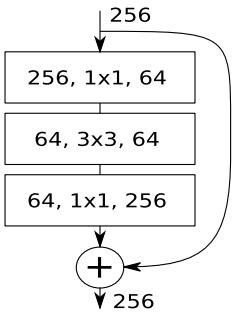

FIGURE 2. The bottleneck residual module [11].

Because of their outstanding performance, different variants of ResNet models have been employed in the field of ear recognition [43], [51], [59]. For our study we consider two ResNet variants, the 50-layer and 101-layer architectures.

We found that an input size of 161 × 257 works the best for ear images from the EarVN1.0 dataset.

E. ResNeXt

ResNeXt [47] is a highly modularized CNN architecture that won the 2nd position in the ILSVRC-2016 competition for image classification and localization [50]. ResNeXt follows the design simplicity of VGGNet and ResNet for constructing deep networks. First, by stacking multiple layers or building blocks of similar architecture having the same number of channels and filter sizes. Second, when the spatial dimension is reduced by a factor of 2, the number of channels is doubled. ResNeXt also adopts the split-transform-merge strategy from the Inception module, but employs an identical set of transformations in all paths; thus, allowing the number of paths to be easily extended and investigated as an independent hyperparameter. The size of the set of transformations is referred to as cardinality, which is argued to be an important dimension to improve the network's performance.

Figure 3 shows the ResNeXt building block with a cardinality of 32. The module performs a similar set of transformations in all paths whose outputs are aggregated by summation. The network is constructed by a stack of ResNeXt blocks along with other conventional convolution and pooling layers. The reader is referred to [47] for a detailed description of the architecture.

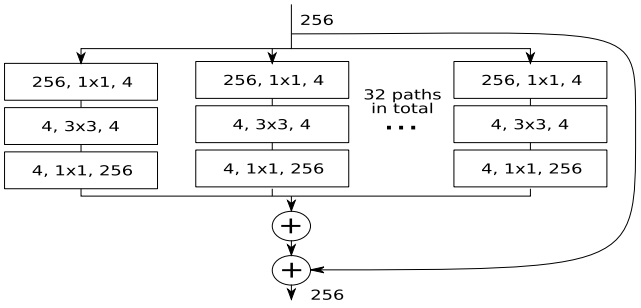

FIGURE 3. The ResNeXt module with cardinality of 32 [47].

For our experiments we implement two ResNeXt variants, the 50-layer and 101-layer architectures. Similar to their ResNet counterparts, original ResNeXt models use an RGB-input of 224 × 224. However, we use an input size of 161 × 257 similar to their counterparts ResNet models as it achieves the best recognition performance. To our knowledge, this work is the first to embrace ResNeXt models for ear recognition experiments.

IV. TRANSFERABILITY OF DEEP FEATURES

Deep CNNs trained on large image datasets exhibit high degrees of transferability of their learned features across different vision tasks and datasets. The transferability becomes more effective as the similarity between the pretraining and target tasks increases. However, transferring the learned features even from a distant task has been proven to be better than learning them from scratch on the target dataset [60]–[67]. Therefore, to address the unconstrained ear recognition problem on the EarVN1.0 dataset we leverage the feature transferability of the top-performing CNN architectures trained for image recognition using the ImageNet dataset [48]. The learned features provide a strong starting point for building robust recognition models [15], [68], [69], even though the new tasks may have different number of classes and images. In this paper we study the feature transferability of pretrained deep networks under two scenarios: feature extraction and fine-tuning. We also explore the power of ensemble learning wherein we combine the prediction of multiple fine-tuned networks to boost the overall recognition performance. The next subsections cover more details on each of the considered scenarios.

A. FEATURE EXTRACTION

Feature extraction is a common transfer learning method to exploit the learned representations from previously trained deep CNNs. It is an effective approach to overcome the computational costs required to train deep networks from scratch and to exploit the set of discriminative filters learned by the network during initial training. The pretrained filters can be utilized to extract interesting features from new image sets and for different visual tasks. The extracted features are then used to train a fully connected network or a standalone classifier.

Typically, a CNN architecture consists of two main parts: the first set of convolution and pooling layers which is referred to as the convolutional base, and a set of fully connected layers on top to perform the classification task. Under the feature extraction scenario, the pretrained CNN architecture and the learned filters are retained. Extracting the features is accomplished by applying the learned filters on new image sets considering the generality of the learned filters for other related vision tasks [70], [71]. This approach has been investigated in several studies, where the images are fed into a pretrained CNN and forward propagated through the network up to a certain layer, and the features extracted by that specific layer are used to train a densely connected network on top [15], [60], [72] or traditional classifiers [70], [73].

For our experiments, we consider the features extracted by the convolutional base of each CNN architecture. We employ three fully connected layers on top of the convolutional base that act as multilayer perceptrons (MLPs) for classification. The first two layers consist of 2048 neurons each, whereas the last layer has 164, matching the number of subjects in the EarVN1.0 dataset. It is worthy mentioning that extending the convolutional base by adding layers on top has two advantages than using standalone classifiers. First, it allows the networks to be trained in an end-to-end manner. Second, it enables applying data augmentation techniques that can lead to better generalization of the trained networks.

B. FINE-TUNING

Fine tuning is another effective transfer learning technique to utilize the capabilities of pretrained CNNs. It involves

performing a network surgery and modifying the CNN architecture in order to achieve better performance. The process starts with removing the final set of densely connected layers (e.g. the network head) from a pretrained CNN, and attaching a new head with a set of randomly initialized densely connected layers. The learned weights for all layers below the network head are kept fixed and only the new head is trained. Once the head has been trained we unfreeze all the layers and continue training the entire network until convergence. This procedure minimizes the domain divergence by identifying discriminative features and progressively adapts them to suit the target recognition task.

Fine-tuning deep CNNs initially trained on the ImageNet dataset has become the de facto standard for building robust and high performing recognition models for vision and related domains [74], including ear recognition [43], [45], [46]. In order to apply fine-tuning we follow a two-step procedure. First, we replace the pretrained head with a new one, which can be a single layer as in all convolutional networks (e.g. Inception, ResNet and ResNeXt) or three fully connected as for AlexNet and VGGNet variants. Additionally, we simplify the newly added head for AlexNet and VGGNet models by using half the number of neurons in the fist two layers. When training the newly attached head it is necessary to freeze all the layers below it in order not to destroy the pre-learned filters. Second, once the head has been trained we unfreeze all layers and jointly fine-tune them along with the trained head.

C. ENSEMBLE LEARNING

Ensemble learning refers to the process of averaging the prediction of multiple deep models trained independently for the same task. Ensembling of deep models is a powerful approach for improving the recognition performance of a single model. In fact, state-of-the-art results in various recognition competitions and particularly the ImageNet challenge [50] are achieved through ensembles of deep networks. Furthermore, the best ear recognition performance in several conducted studies are achieved through ensembles of deep models [43], [46], [75].

To construct the deep ensembles, we independently fine-tune 10 models from each CNN architecture. We sort the models in a descending order based on their accuracies. We build ensembles by picking up the best performing model among the 10 models and continue to gradually combine more models. Each model will produce a 164-dimensional vector of class probabilities for each test image. The probability vectors are then averaged across the number of networks in the deep ensemble. The final prediction is assigned to the subject with the highest probability value in the averaged vector. The entire process is depicted in Figure 4. In Section VI we show how this approach can lead to an improved recognition performance for each of the considered CNN architecture.

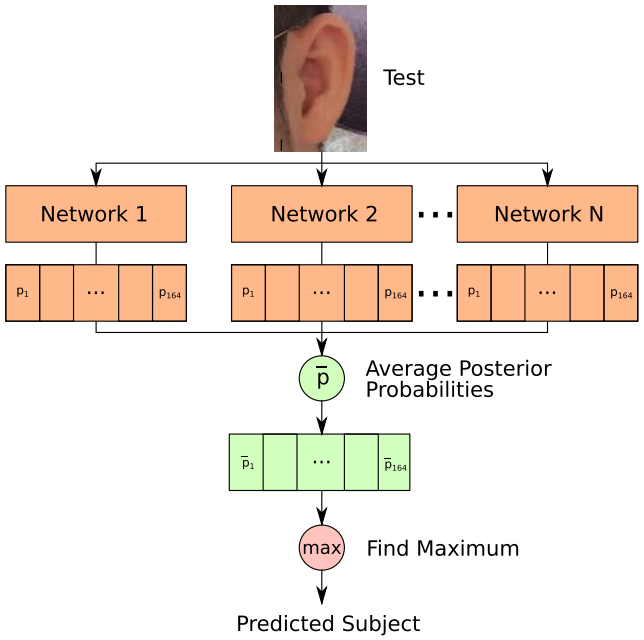

FIGURE 4. The process of ensemble prediction with multiple fine-tuned CNN models. The test image is passed to each model, which produces an independent vector of class probabilities. The obtained probability vectors are averaged across the ensemble members. The final prediction is assigned to the class with the highest probability of the averaged vector.

V. EXPERIMENTAL SETUP

To conduct the recognition experiments, we split the EarVn1.0 dataset into two disjoint sets, training and test, holding 60% and 40% of ear images, respectively. The training set is used to fine tune the weights of the different networks, whereas the test set is used for evaluation and reporting the results.

All the networks are trained using the back-propagation algorithm [76] and the LAMB optimizer [21] on a cross-entropy loss using momentum [77] with a decay of 0.9. We observe that a weight decay of 0.0001 leads to less over-fitting and we apply it to all of our experiments. The networks are trained on a desktop PC with Intel(R) Core(TM) i7-3770 CPU, 8 MB RAM and Nvidia GTX 1080 until convergence.

The learning rate is scheduled to have an initial value of 0.001 and it is decreased depending on the learning strategy to 0.00001. For feature extraction (FE) and feature extraction plus batch normalization (FE + BN) strategies, we train the networks for 150 epochs. For the fine-tuning strategies we train for different number of epochs. In case fine-tuning with square-sized inputs we train AlexNet, VGG16 and VGG19 for 220 epochs, and the other networks for 100 epochs. The InceptionV3 and all residual networks need fewer epochs to converge. One reason is the lack of incremental training and the very good convergence behavior. However, fine tuning with custom-sized inputs needs more epochs than with square-sized inputs in some cases. For instance, AlexNet needs 300 epochs to converge while

VGG16 and VGG19 need 400 epochs. The InceptionV3 and all residual networks need 100 epochs to reach convergence.

The change of the input size makes it necessary to do larger adjustments of the pretrained models and the filters have to be changed to a larger extend to suit the new resolution. However, the benefit of the longer training time is only around 1% and barely noticeable. The custom input sizes are chosen based on the average aspect ratio of the EarVN1.0 dataset. In addition we match the input size and filter sizes in the particular network so that convolution operations fit each intermediate activation flawlessly without skipping any pixels.

A. EAR DATASET

The recently released EarVN1.0 dataset is to be considered one of the largest ear datasets collected under unconstrained conditions [36]. A total of 28,412 ear images for 164 subjects are collected from both genders and for the left and right ears. The images are in RGB format and have variable spatial resolution between 15×15 and 200×200 pixels. The ear images are cropped from facial profiles captured under uncontrolled settings with different acquisition devices and exhibit large variations in scale, viewing angles, illumination, contrast, and small background artifacts. To highlight the difficulty of the EarVN1.0 dataset, Figure 5 presents sample ear images for three subjects.

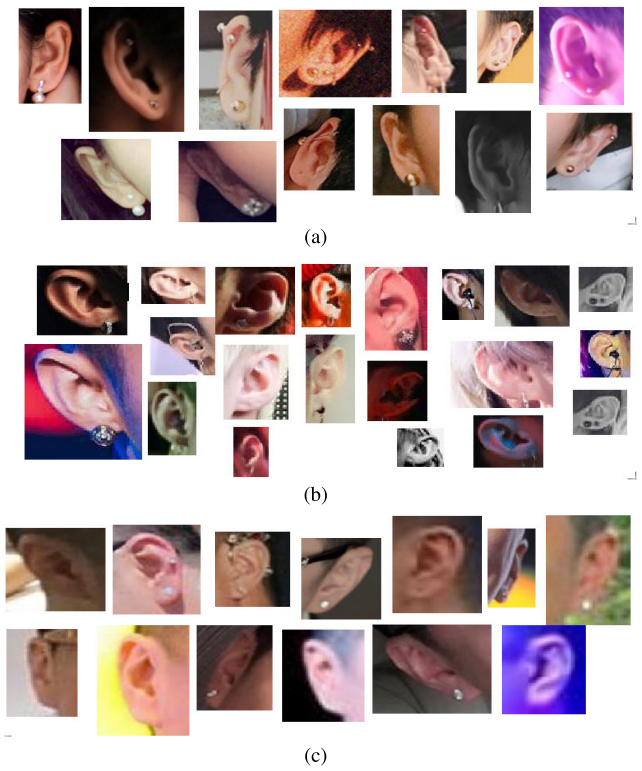

FIGURE 5. Sample ear images for three different subjects from the EarVN1.0 dataset. The images exhibit wide range of variations in illumination, head poses, scale, image resolution, occlusions, and background effects.

B. DATA AUGMENTATION

Training deep CNNs requires enormous corpora of annotated examples to learn from, and to extract more class-specific features accurately. When the large-scale training data is not available, deep CNNs tend to overfit on small datasets. An effective approach to address the issue is to augment the training examples via a set of label-preserving transformations, which perturb the training examples by slightly changing their appearance before feeding them into the networks for training. For our experiments, we apply a wide range of augmentation techniques to introduce appearance variations that reflect the situations found in real-world images. The goal of applying data augmentation is two-fold: (1) this exposes the networks to various aspects of the training images which increases the network generalization, and (2) helps the networks to learn more robust features as a result of constantly seeing altered versions of the input images. The augmentation techniques are carried out on the fly during training to avoid the extra memory space required for storing the images.

The set of image perturbations are performed randomly during the training process. After loading an image from disk, we rescale it to the approximate target size, while keeping the original aspect ratio intact. In order to match the target size we apply appropriate padding with the average color of the ImageNet dataset. Subsequently, random rotation is performed between -15 and 15 degrees. The choice of this number turns out to be quite relevant. Lower angles lead to more over-fitting but higher angles tend to decrease the recognition performance in general. Shearing is not performed because we found it to have a negative impact on training. Then, the image is randomly cropped and resized to the target input size. We blur the image with a probability of 20% and add Gaussian noise to it. Also, brightness, contrast, saturation and hue are randomly changed. Finally, the image is flipped horizontally with a probability of 50%. Then it is normalized according to the ImageNet dataset.

C. EVALUATION METRICS

The recognition performance is evaluated using three quantitative measures and is visualized by plotting the Cumulative Match Characteristics (CMC) curves for each experiment. A brief description for each of the metrics is given below.

- Cumulative Match Characteristics (CMC) curve: is a rank-based metric showing the probability that the model will return the correct identity within the top k ranks ($k \leq N$) where N is the number of subjects in the gallery.
- Rank-1 recognition rate (R1): refers to the percentage of probe images for which the correct identity is found as a top match from the gallery.
- Rank-5 recognition rate (R5): is the percentage of probe images for which the correct identity is found within the top five matches from the gallery.
- Area under the CMC curve (AUC): is an objective measure for recognition performance identical to the widely

used area under the Receiver Operating Characteristic curve.

VI. EXPERIMENTS AND RESULTS

This section presents the results of our experimental study and analysis. First, we report the comparative recognition performance for the deep models under the different transfer learning strategies. Second, we visualize the extracted features under each strategy using the t-SNE algorithm and highlight the main differences. Finally, we compare the training time and model size of each CNN architecture along with some distinguishing hyperparameters.

A. COMPARATIVE ANALYSIS

The performance of eight different deep CNN architectures is evaluated on the EarVN1.0 dataset through three sets of recognition experiments. The first set of experiments is to determine how good the pretrained ImageNet models perform in representing ear images from the EarVN1.0 dataset. To this end, we consider the feature extraction strategy wherein we attach the fully connected layers at the top of the convolutional part for each pretrained network. We then assess the impact of training all batch normalization layers along with the fully connected layers. The second set of experiments evaluates the proposed two-step fine-tuning strategy using ear images with square-sized inputs identical to the inputs to the original CNN architectures. We also study the performance of the different networks when fine-tuned with custom-sized inputs intended to preserve the aspect ratio of ear images. The third set of experiments is to measure the performance gain when several deep models are combined in an ensemble. The results from all experiments are presented in Table 1. The R1, R5 and AUC values attained by the best performing models for each learning strategy are written in bold. To analyze the overall recognition performance across the different ranks, the CMC curves are also provided for each experiment in Figures 6, 7 and 10.

Our first set of experiments on feature extraction is based on the hypothesis that the pretrained models have already learned a set of generic filters. This is intended to examine how discriminative the learned convolutional filters are in extracting robust ear image features from the EarVN1.0 dataset. Also, it allows us to obtain a baseline for benchmarking other learning strategies and our new experiments. We use the same initialization in all networks for the newly added layers and train them until convergence. Intriguingly, the obtained results show that the extracted features can not discriminate the ear images of the different subjects from the EarVN1.0 dataset. A logical interpretation is that the spatial hierarchies of features learned for generic objects during the initial training on the ImageNet dataset can hardly be found in ear images. Under this strategy AlexNet obtains the highest recognition performance with a rank-1 accuracy of 30%. The other models perform in a similar manner as reported in Table 1 and visualized in Figure 6 (a). However, the obtained results are far from being satisfactory and indicate the need for adjusting more layers to obtain relevant ear features and a better recognition performance.

TABLE 1. The recognition results obtained with different CNN models and various learning strategies. The results are given in percentages where the top three values for each learning strategy are highlighted in bold.

Strategy	Model	Evaluation Metrics		
		R1	R5	AUC
Feature Extraction	AlexNet	**29.83**	**55.98**	**92.49**
	VGG16	24.67	49.67	90.76
	VGG19	13.44	32.71	84.53
	InceptionV3	21.51	45.84	89.86
	ResNet50	28.52	55.12	92.08
	ResNeXt50	21.55	44.26	89.40
	ResNet101	18.23	42.46	88.58
	ResNeXt101	26.85	52.06	91.09
Feature Extraction + Batch Normalization	AlexNet	55.67	79.30	96.72
	VGG16	75.15	90.74	98.32
	VGG19	75.86	91.52	98.33
	InceptionV3	81.85	94.29	98.73
	ResNet50	83.06	94.64	98.76
	ResNeXt50	84.43	95.43	98.85
	ResNet101	86.23	95.79	98.90
	ResNeXt101	**89.13**	**96.68**	**98.99**
Fine Tuning with Square-sized Inputs	AlexNet	78.93	92.86	98.55
	VGG16	88.79	96.52	98.97
	VGG19	89.93	96.89	99.01
	InceptionV3	**90.40**	**97.35**	**99.06**
	ResNet50	89.35	96.76	99.00
	ResNeXt50	89.27	96.52	98.97
	ResNet101	89.15	96.76	99.02
	ResNeXt101	**90.40**	97.04	99.02
Fine Tuning with Custom-sized Inputs	AlexNet	78.62	92.62	98.54
	VGG16	88.73	96.81	98.99
	VGG19	89.34	96.73	99.01
	InceptionV3	90.67	97.18	99.01
	ResNet50	91.84	97.54	99.11
	ResNeXt50	91.83	97.72	99.11
	ResNet101	92.85	97.95	99.16
	ResNeXt101	**93.45**	**98.42**	**99.18**

In order to keep the parameter space small, we propose to train all the batch normalization layers along with the newly added fully connected layers, which we refer to as feature extraction + batch normalization (FE + BN). Surprisingly, we observe a drastic change in the evaluation metrics compared with the first learning strategy. Applying batch normalization is a very effective approach for stabilizing the training and accelerates the convergence process. It normalizes the activations of an input volume before feeding them into the next layer, which reduces the internal covariate shift problem. The conducted experiments indicate that training the batch normalization layers along with the fully connected ones improves the generalization ability of the networks and considerably boost the recognition accuracy. When considering the evaluation metrics in Table 1, we notice that, as the network becomes deeper the recognition performance keeps improving as a result of learning more discriminative image representations. The performance of the AlexNet model raised from 29.8% to 55.6%, whereas the best performing model is the ResNeXt101 model, which achieves rank-1 accuracy of 89% on the test set. With this strategy all the networks attain a reasonable recognition accuracy with a clear advantage for deeper networks. This is also consistent across the different recognition ranks as can be seen from the CMC curves in Figure 6 (b).

Next, we employ a two-step fine-tuning strategy for the networks that have more than one fully connected layer, i.e., AlexNet, VGG16, and VGG19. First, we train the

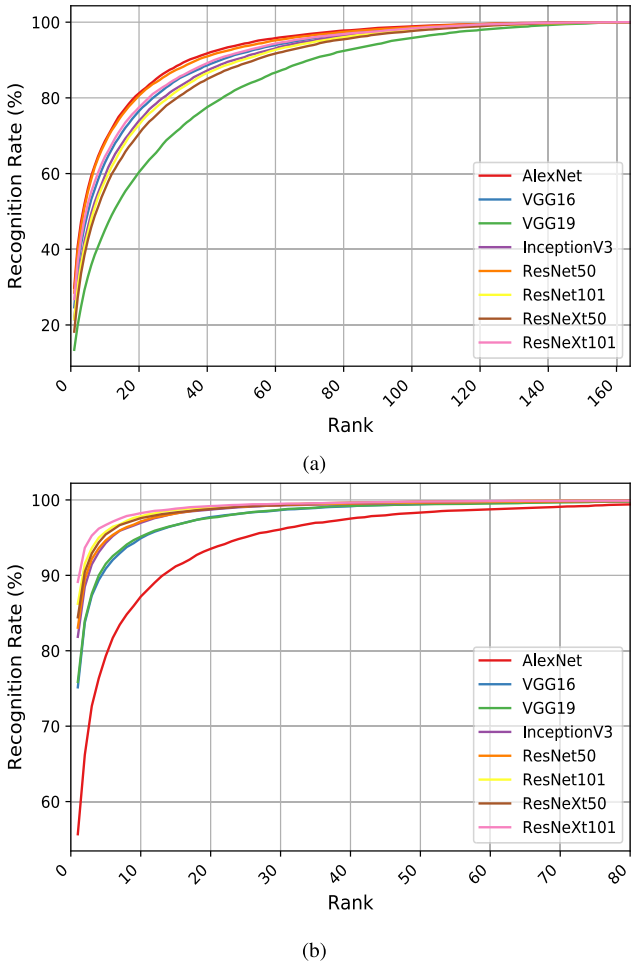

FIGURE 6. The CMC curves comparing the performance of the different CNN models when: (a) applying the feature extraction strategy, and (b) applying the feature extraction and batch normalization strategy.

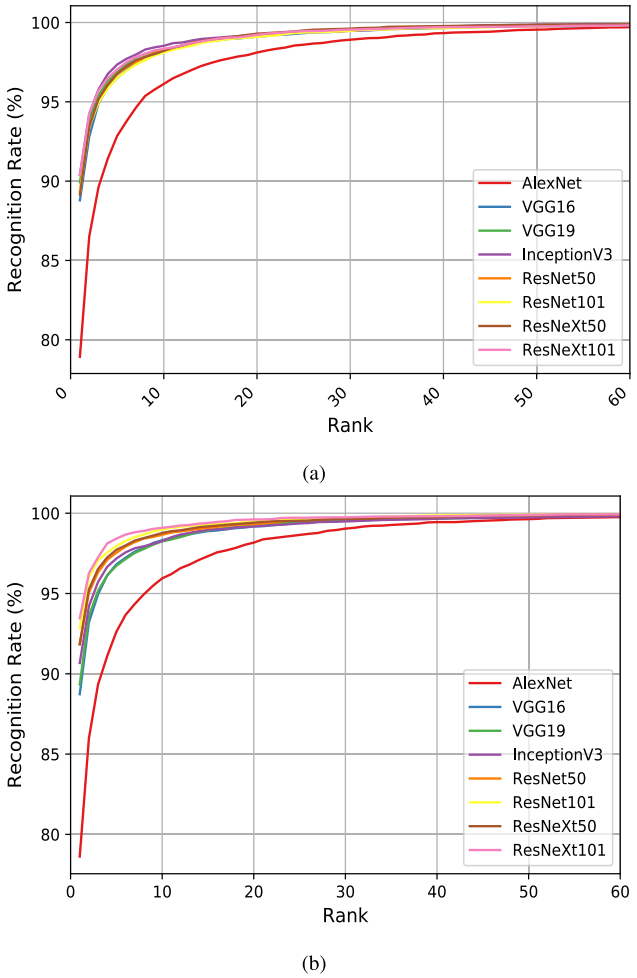

FIGURE 7. The CMC curves comparing the performance of the different CNN models when the models are fine-tuned using: (a) square-sized input images and (b) custom-sized inputs suitable for each model.

networks to adjust the weights of the newly added fully connected layers until convergence using the training set, meanwhile all the convolutional layers are kept unchanged in order to avoid destroying the pre-learned representations. Second, we allow the weights of all layers including the fine-tuned fully connected ones to be adjusted in a second round of tuning using the exact 60% of training images. These experiments are conducted using square-sized inputs such as 224×224, and 299×299, as used by the original CNN architectures. All images are preprocessed by resizing them to the required input size. Again, the reported results in Table 1 indicate that deeper models attain an improved recognition performance. So, under this learning strategy, the AlexNet model achieves rank-1 recognition accuracy of 78.93%, which represents a remarkable improvement of 23% over the FE + BN learning strategy. Also, the VGG16 and VGG19 models achieve better performance with relative improvements of approximately 14%. Interestingly, the InceptionV3 model also shows an improved performance with an increase of 9% and hits 90.40% recognition accuracy. The ResNeXt101 model achieves similar accuracy to InceptionV3 but with a slight improvement in the higher ranks as presented in Figure 7 (a). The other models of ResNet50, ResNeXt50 and ResNet101 attain above 89% accuracy and show approximately identical performance across all evaluation metrics.

We perform a similar set of the fine-tuning experiments using a custom-sized input that suits the target CNN architecture and is intended to preserve the aspect ratio of ear images from the EarVN1.0 dataset. The obtained results from the conducted experiments are presented in Table 1 and visualized in Figure 7 (b). We notice an improvement in the performance metrics from 2% to 3% for the residual networks (i.e., ResNet and ResNeXt) as compared to their counterparts when using fixed-size inputs. However, no significant difference in the performance metrics is observed for AlexNet, VGGNet and InceptionV3 models. The results indicate that the fine-tuned deeper models such as ResNet101 and ResNeXt101 perform the best as more discriminative features are learned. The best overall performance is achieved by ResNeXt101 with a rank-1 recognition accuracy of 93.45%,

which represents a relative improvement of 3% over its fine-tuned counterpart using square-sized inputs. Moreover, we observe that the ResNeXt101 model is also the top performer over all ranks and metrics among the evaluated networks. These results indicate that keeping the aspect ratio of the ear images, which are rectangular in nature, is indeed beneficial and helps to improve the performance to a certain degree. This indicates that fine-tuning is and effective transfer learning technique to adapt the learned features from a knowledge domain to a target domain. More specifically, fine-tuning deeper networks generates better features than shallower networks. From the conducted experiments and the obtained results, we can infer that the fine-tuned networks have learned more discriminative and ear-specific features capable of discriminating between the different subjects from the EarVN1.0 dataset.

Until now, we measure and compare the performance of single fine-tuned models. In our third set of experiments we build deep ensembles to further improve the recognition accuracy. A deep ensemble combines the predictions of several deep networks where the final prediction is computed by averaging the posterior probabilities obtained by the softmax layers for all the ensemble's members. To this end, we train 10 networks from each architecture using similar initialization and learning schedules. Even though we use the same training split for training all networks, they only differ in the random process of shuffling the training images and the different augmentation steps applied per batch. The networks are trained independently and due to their stochastic nature each network learns variations of filters and patterns. These variations can be exploited to improve the recognition accuracy when combining the prediction of multiple networks. Figure 8 shows the box plots that highlight the variance between the 10 models from each network against the rank-1 accuracy. We can see that AlexNet models have more variations, whereas the other fine-tuned networks show a slight variance in performance between the different models.

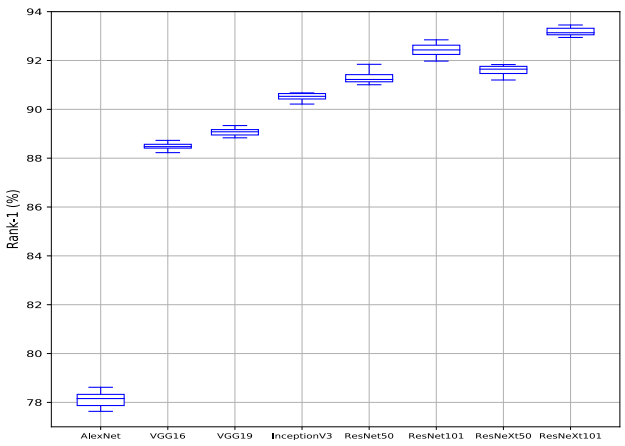

FIGURE 8. The box plots illustrate the variance in recognition performance for 10 different models from each CNN architecture.

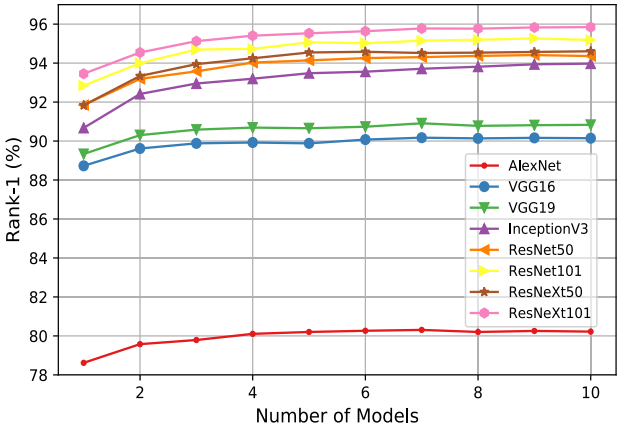

FIGURE 9. The change in rank-1 recognition accuracy with different number of models in the ensemble. A noticeable improvement in performance is observed for all CNN architectures when increasing the ensemble's members.

We start by constructing an ensemble of n models where n ranges from 1 to 10. Figure 9 compares the rank-1 recognition accuracy and the number of models in the ensemble. In general, we observe improvements of the recognition performance for all networks when using the ensemble prediction. More specifically, we notice that the rank-1 accuracy keeps increasing as the ensemble members increase for AlexNet, VGG16 and VGG19 up to the sixth model, where they achieve their best performance of 80.31%, 90.17% and 90.91%, respectively. While the performance of InceptionV3 and the residual networks keeps improving when adding more models to the ensemble up to approximately the 10th model. The best rank-1 accuracies achieved by ensembles of 10 models from InceptionV3, ResNet50, ResNet101, ResNeXt50 and ResNeXt101 are 93.97%, 94.41%, 95.27%, 94.61%, 95.85%, respectively. Figure 10 visualizes the CMC curves for the 10-models ensembles and the relative recognition performances. Overall, the best recognition performance is achieved by the ResNeXt101 model when comparing the single model prediction as well as ensemble prediction with rank-1 rates of 93.45% and 95.85%, respectively. These are the best results achieved by our models and the first to be reported on the EarVN1.0 dataset.

B. FEATURE VISUALIZATION

In this section we examine the networks' visual knowledge and the impact of our adaptation strategies on the obtained features. The t-SNE dimensionality reduction and visualization technique [78] is used to explore and visualize the features. We consider the activations extracted from the penultimate layer before the softmax layer of the ResNeXt101 model as it is the best performing model. Since the extracted features are 2048-dimensional vectors, t-SNE embeds the vectors into a 2D space and at the same time preserves the local neighborhood of the feature vectors. In order to make the feature visualizations clearer and avoid clutter

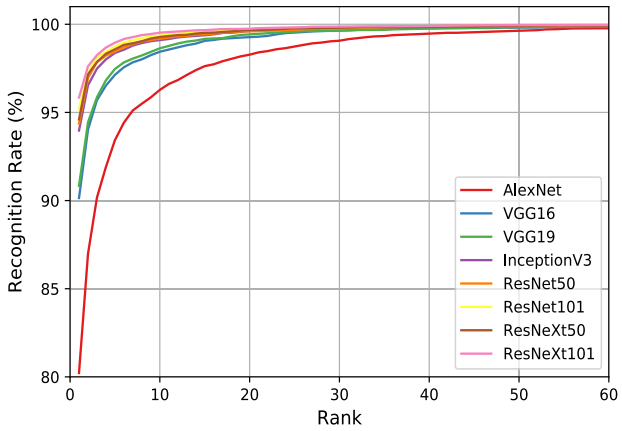

FIGURE 10. The CMC curves show the comparative recognition performance of the deep ensembles of 10 models from each architectures. The best recognition results across all ranks are achieved by an ensemble of 10 ResNeXt101 models.

in the figures, we visualize the learned features for only 50 subjects. The feature visualizations help to interpret the performance differences between the various ResNeXt101 models obtained by the different learning strategies.

First, we consider the ResNeXt101 model pretrained on ImageNet without tuning any of the weights. The test images are propagated forward through the pretrained network and the activations from the penultimate layer are extracted for each image. Then, the resulting feature vectors are projected onto a 2D space using the t-SNE algorithm for visualization. The extracted features from ear images of the same subject are expected to be close in the feature space. Figure 11 illustrates the 2D map of the features extracted by the the ResNeXt101 model pretrained on ImageNet. The features do not show any clustering behavior, which indicates that the extracted features are not discriminative enough to cluster the individuals correctly. This also explains the weak recognition results in Table 1.

Second, we consider the fine-tuned ResNeXt101 model in order to investigate the effect of fine-tuning on the resulting features. As can be seen in Figure 12, a significant improvement in the visualization is achieved. The extracted features are clearly clustered into groups, and each group represents a different subject. The obtained visualization shows that fine-tuning the pretrained ImageNet models adapts the model to learn more ear-specific and discriminative features.

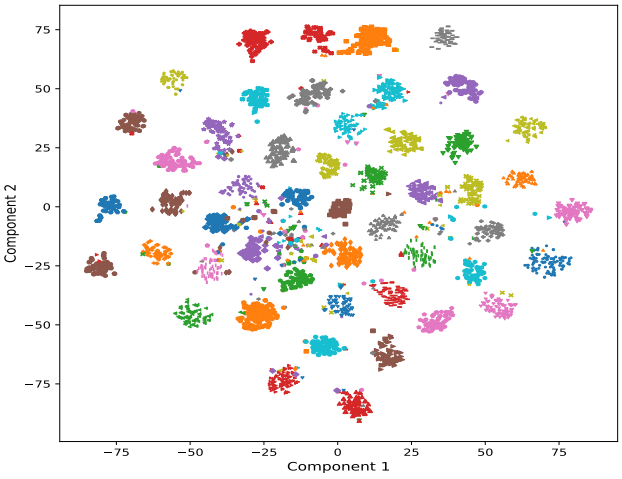

FIGURE 12. The t-SNE visualization of features extracted by the fine-tuned ResNext101 model using square-sized input images. Best viewed in color.

Third, for the sake of comparability and to interpret the improvement in recognition accuracy, we visualize the features extracted by the fine-tuned ResNeXt101 model when using custom-sized input images. Interestingly, the extracted features form more visible clusters as shown in Figure 13, where we can clearly see 50 clusters, one for each subject.

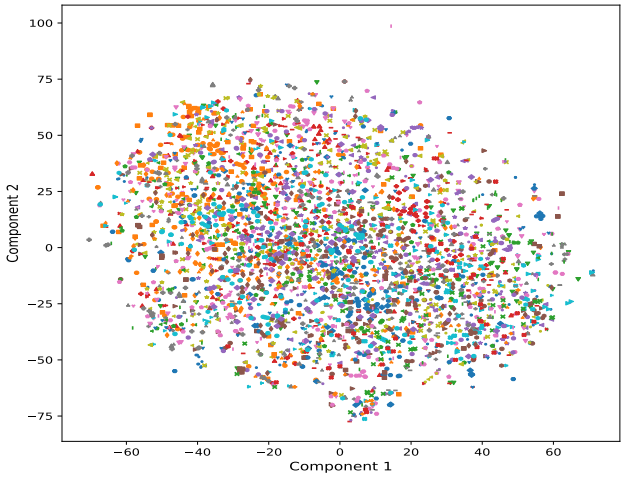

FIGURE 11. The t-SNE visualization of features extracted by the ResNext101 model pretrained on ImageNet without fine-tuning. Best viewed in color.

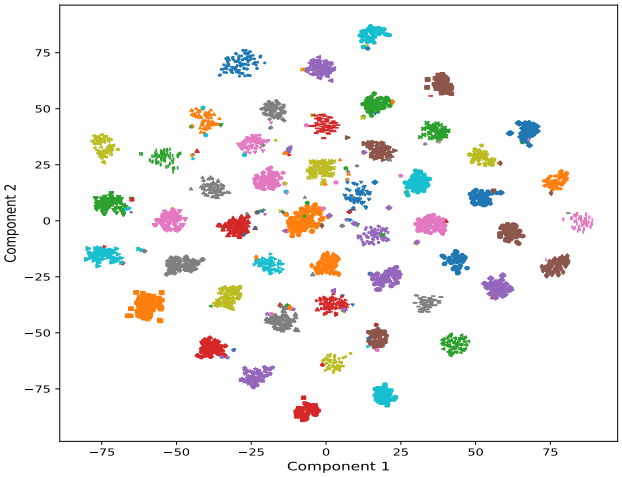

FIGURE 13. The t-SNE visualization of features extracted by the fine-tuned ResNext101 model using custom-sized input images. Best viewed in color.

TABLE 2. A comparison of the distinguishing characteristics of the CNN architectures investigated in our experiments.

Architecture	Input size		Model size (MB)		Trainable params (m)		Convolution output		Feature size		Training time(s)/epoch	
	Default	Ours	Default	Ours	Default	Ours	Default	Ours	Default	Ours	Default	Ours
AlexNet	227×227	201×297	220.2	74.8	57.7	19.6	9216	1536	4096	2048	72	82
VGG16	224×224	192×288	515.0	169.5	135	44.5	25088	3072	4096	2048	196	174
VGG19	224×224	192×288	535.3	189.8	140.3	49.8	25088	3072	4096	2048	200	231
InceptionV3	299×299	171×235	84.4	84.4	22.1	22.1	2048	2048	2048	2048	150	76
ResNet50	224×224	161×257	90.9	90.9	23.8	23.8	2048	2048	2048	2048	106	103
ResNet101	224×224	161×257	163.4	163.4	42.8	42.8	2048	2048	2048	2048	179	175
ResNeXt50	224×224	161×257	88.9	88.9	23.3	23.3	2048	2048	2048	2048	193	190
ResNeXt101	224×224	161×257	332.1	332.1	87.1	87.1	2048	2048	2048	2048	443	441

Also, the number of outliers that can be observed in the middle of Figure 12 are reduced and the features are clustered more closely. Generally, the feature visualizations highlight the success of the proposed fine-tuning strategies and give more insights into how the networks see ear images and how they extract semantic meaning associated with each individual.

C. COMPLEXITY ANALYSIS

In this section we highlight the different characteristics of the deep networks investigated in our experiments. Table 2 summarizes the important factors to be considered when working with deep CNNs such as the required memory space to store a model, number of trainable parameters, training time, the size of the final feature vector, and the input size for each architecture. The default values for each architecture represent the pretrained ImageNet models with the same input resolution used when these networks were trained. The only change is in the number of trainable parameters in the newly added softmax layer. However, the values for 'Ours' represent the same CNN models when using custom-sized inputs and reducing the number of neurons in the fully connected layers by half as for AlexNet and VGGNet models.

Note in Table 2 a significant reduction in the model size and number of trainable parameters for our models, compared to the original architectures and more specifically the AlexNet VGGNet models, which is attributed to several factors. First, the default models were proposed to classify the ImageNet dataset which has 1000 classes, however, the EarVN1.0 dataset has only 164 distinct subjects leading to less parameters in the last layer. Second, we use half the number of neurons in the fully connected layers for AlexNet and VGGNet architectures, which also helps to combat overfitting. Third, due to the small size of ear images compared with the ImageNet data, we found it more efficient to use a small input resolution for most CNN architectures. For instance, the InceptionV3 model has the lowest computational cost among all models and uses input resolutions with less than half the number of pixels, i.e., $171 \times 235 \times 3 = 120555$ instead of $299 \times 299 \times 3 = 268203$, and still obtains better performance. Generally, the modified fine-tuned networks not only improve the recognition performance, but also are more computationally efficient.

VII. CONCLUSION

In this paper, we reported the results of the first experimental study of ear recognition on the EarVN1.0 dataset, which is considered to be one of the largest datasets collected under unconstrained conditions. Inspired by their impressive recognition performance in various vision tasks, we employed eight different deep CNN architectures and effective transfer learning strategies to obtain the best recognition performance.

We examined the generalization ability of deep CNNs as feature extractors. The obtained results showed unsatisfactory recognition performance. Therefore, we proposed a two-step fine-tuning strategy for networks with more than one fully connected layer. First, we trained the newly added layers until convergence and then fine-tuned the entire network in a second round. The obtained results showed significant improvements for all networks with a rank-1 recognition accuracy above 90%. We also explored training the networks with custom-sized inputs intended to preserve the aspect ratio of ear images and achieved better performance. A single ResNeXt101 model achieved a rank-1 recognition accuracy of 93.45% representing a relative accuracy improvement above 3%.

To improve the recognition performance of single models we investigated the effectiveness of deep ensembles. To this end, we independently trained 10 models from each CNN architecture and built ensembles using n models where n ranged from 1 to 10. The obtained results showed that deep ensembles achieved an improved recognition performance with a relative improvement above 2% over single models. An ensemble of ResNeXt101 models achieved the best rank-1 recognition accuracy of 95.85%. Finally, we applied the t-SNE algorithm to explore and visualize the learned features. The provided visualizations showed well-separated clusters of ear images for the different individuals, which indicated that the extracted features by our proposed learning strategies are more discriminative.

REFERENCES

[1] H. Nejati, L. Zhang, T. Sim, E. Martinez-Marroquin, and G. Dong, "Wonder ears: Identification of identical twins from ear images," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, 2012, pp. 1201–1204.

[2] J. Unar, W. C. Seng, and A. Abbasi, "A review of biometric technology along with trends and prospects," *Pattern Recognit.*, vol. 47, no. 8, pp. 2673–2688, 2014.

[3] Z. Wang, J. Yang, and Y. Zhu, "Review of ear biometrics," in *Archives of Computational Methods in Engineering*. Springer, Nov. 2019, doi: 10.1007/s11831-019-09376-2.

[4] M. Oravec, J. Pavlovicova, D. Sopiak, V. Jirka, M. Loderer, L. Lehota, M. Vodicka, M. Fackovec, M. Mihalik, M. Tomik, and J. Gerat, "Mobile ear recognition application," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, May 2016, pp. 1–4.
[5] A. Poosarala and R. Jayashree, "Uniform classifier for biometric ear and retina authentication using smartphone application," in *Proc. 2nd Int. Conf. Vis., Image Signal Process.*, 2018, pp. 1–5.
[6] S. A. Bargal, A. Welles, C. R. Chan, S. Howes, S. Sclaroff, E. Ragan, C. Johnson, and C. Gill, "Image-based ear biometric smartphone app for patient identification in field settings," in *Proc. VISAPP*, 2015, pp. 171–179.
[7] S. Barra, M. De Marsico, M. Nappi, and D. Riccio, "Unconstrained ear processing: What is possible and what must be done," in *Proc. Signal Image Process. Biometrics*, 2014, pp. 129–190.
[8] Ž. Emeršič, V. Štruc, and P. Peer, "Ear recognition: More than a survey," *Neurocomputing*, vol. 255, pp. 26–39, Sep. 2017.
[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.
[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
[13] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[17] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.
[18] S. Minaee, A. Abdolrashidi, H. Su, M. Bennamoun, and D. Zhang, "Biometric recognition using deep learning: A survey," 2019, *arXiv:1912.00271*. [Online]. Available: http://arxiv.org/abs/1912.00271
[19] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[20] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, p. 9, 2016.
[21] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, and C.-J. Hsieh, "Large batch optimization for deep learning: Training bert in 76 minutes," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.
[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.
[23] M. Jahrens and T. Martinetz, "Solving Raven's progressive matrices with multi-layer relation networks," 2020, *arXiv:2003.11608*. [Online]. Available: http://arxiv.org/abs/2003.11608
[24] A. Benzaoui, A. Hadid, and A. Boukrouche, "Ear biometric recognition using local texture descriptors," *J. Electron. Imag.*, vol. 23, no. 5, Sep. 2014, Art. no. 053008.
[25] A. Benzaoui, I. Adjabi, and A. Boukrouche, "Experiments and improvements of ear recognition based on local texture descriptors," *Opt. Eng.*, vol. 56, no. 4, Apr. 2017, Art. no. 043109.
[26] H. A. Alshazly, M. Hassaballah, M. Ahmed, and A. A. Ali, "Ear biometric recognition using gradient-based feature descriptors," in *Proc. 4th Int. Conf. Adv. Intell. Syst. Inform.*, 2018, pp. 435–445.
[27] D. P. Chowdhury, S. Bakshi, G. Guo, and P. K. Sa, "On applicability of tunable filter bank based feature for ear biometrics: A study from constrained to unconstrained," *J. Med. Syst.*, vol. 42, no. 1, p. 11, Jan. 2018.
[28] J. D. Bustard and M. S. Nixon, "Toward unconstrained ear recognition from two-dimensional images," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 40, no. 3, pp. 486–494, May 2010.
[29] M. Hassaballah, H. A. Alshazly, and A. A. Ali, "Ear recognition using local binary patterns: A comparative experimental study," *Expert Syst. Appl.*, vol. 118, pp. 182–200, Mar. 2019.
[30] D. Frejlichowski and N. Tyszkiewicz, "The West Pomeranian University of Technology ear database—A tool for testing biometric algorithms," in *Proc. Int. Conf. Image Anal. Recognit.*, 2010, pp. 227–234.
[31] Ž. Emeršič *et al.*, "Evaluation and analysis of ear recognition models: Performance, complexity and resource requirements," *Neural Comput. Appl.*, May 2018, doi: 10.1007/s00521-018-3530-1.
[32] Ž. Emeršič, D. Štepec, V. Štruc, P. Peer, A. George, A. Ahmad, E. Omar, T. E. Boult, R. Safdaii, Y. Zhou, S. Zafeiriou, D. Yaman, F. I. Eyiokur, and H. K. Ekenel, "The unconstrained ear recognition challenge," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2017, pp. 715–724.
[33] Y. Zhou and S. Zaferiou, "Deformable models of ears in-the-wild for alignment and recognition," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2017, pp. 626–633.
[34] Y. Zhang and Z. Mu, "Ear detection under uncontrolled conditions with multiple scale faster region-based convolutional neural networks," *Symmetry*, vol. 9, no. 4, p. 53, Apr. 2017.
[35] Y. Zhang, Z. Mu, L. Yuan, C. Yu, and Q. Liu, "USTB-Helloear: A large database of ear images photographed under uncontrolled conditions," in *Proc. Int. Conf. Image Graph.*, 2017, pp. 405–416.
[36] V. T. Hoang, "EarVN1.0: A new large-scale ear images dataset in the wild," *Data Brief*, vol. 27, Dec. 2019, Art. no. 104630.
[37] L. Tian and Z. Mu, "Ear recognition based on deep convolutional network," in *Proc. 9th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, Oct. 2016, pp. 437–441.
[38] T. Ying, W. Shining, and L. Wanxiang, "Human ear recognition based on deep convolutional neural network," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2018, pp. 1830–1835.
[39] Ž. Emeršič, D. Štepec, V. Štruc, and P. Peer, "Training convolutional neural networks with limited training data for ear recognition in the wild," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2017, pp. 988–994.
[40] Y. Khaldi and A. Benzaoui, "A new framework for grayscale ear images recognition using generative adversarial networks under unconstrained conditions," *Evolving Syst.*, May 2020, doi: 10.1007/s12530-020-09346-1.
[41] E. E. Hansley, M. P. Segundo, and S. Sarkar, "Employing fusion of learned and handcrafted features for unconstrained ear recognition," *IET Biometrics*, vol. 7, no. 3, pp. 215–223, May 2018.
[42] H. Alshazly, C. Linse, E. Barth, and T. Martinetz, "Handcrafted versus CNN features for ear recognition," *Symmetry*, vol. 11, no. 12, p. 1493, Dec. 2019.
[43] S. Dodge, J. Mounsef, and L. Karam, "Unconstrained ear recognition using deep neural networks," *IET Biometrics*, vol. 7, no. 3, pp. 207–214, May 2018.
[44] Y. Zhang, Z. Mu, L. Yuan, and C. Yu, "Ear verification under uncontrolled conditions with convolutional neural networks," *IET Biometrics*, vol. 7, no. 3, pp. 185–198, May 2018.
[45] H. Alshazly, C. Linse, E. Barth, and T. Martinetz, "Ensembles of deep learning models and transfer learning for ear recognition," *Sensors*, vol. 19, no. 19, p. 4139, Sep. 2019.
[46] Ž. Emeršič *et al.*, "The unconstrained ear recognition challenge 2019," in *Proc. IEEE Int. Conf. Biometrics*, Jun. 2019, pp. 1–15.
[47] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.
[48] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.
[49] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[50] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[51] F. I. Eyiokur, D. Yaman, and H. K. Ekenel, "Domain adaptation for ear recognition using deep convolutional neural networks," *IET Biometrics*, vol. 7, no. 3, pp. 199–206, May 2018.
[52] A. A. Almisreb, N. Jamil, and N. M. Din, "Utilizing AlexNet deep transfer learning for ear recognition," in *Proc. 4th Int. Conf. Inf. Retr. Knowl. Manage. (CAMP)*, Mar. 2018, pp. 1–5.
[53] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.
[54] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[55] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," 2014, *arXiv:1404.5997*. [Online]. Available: http://arxiv.org/abs/1404.5997
[56] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
[57] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[58] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*. [Online]. Available: http://arxiv.org/abs/1312.4400
[59] D. Štepec, Ž. Emeršič, P. Peer, and V. Štruc, "Constellation-based deep ear recognition," in *Deep Biometrics*. Cham, Switzerland: Springer, 2020, pp. 161–190.
[60] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.
[61] J. Ngiam, D. Peng, V. Vasudevan, S. Kornblith, Q. V. Le, and R. Pang, "Domain adaptive transfer learning with specialist models," 2018, *arXiv:1811.07056*. [Online]. Available: http://arxiv.org/abs/1811.07056
[62] W. Ge and Y. Yu, "Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1086–1095.
[63] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie, "Large scale fine-grained categorization and domain-specific transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4109–4118.
[64] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4068–4076.
[65] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 97–105.
[66] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 36–45.
[67] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (BiT): General visual representation learning," 2019, *arXiv:1912.11370*. [Online]. Available: http://arxiv.org/abs/1912.11370
[68] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1717–1724.
[69] O. A. B. Penatti, K. Nogueira, and J. A. dos Santos, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 44–51.
[70] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 647–655.
[71] L. Hertel, E. Barth, T. Käster, and T. Martinetz, "Deep convolutional neural networks as generic feature extractors," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–4.
[72] S. Kornblith, J. Shlens, and Q. V. Le, "Do better ImageNet models transfer better?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2661–2671.
[73] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features Off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 806–813.
[74] M. Huh, P. Agrawal, and A. A. Efros, "What makes ImageNet good for transfer learning?" 2016, *arXiv:1608.08614*. [Online]. Available: http://arxiv.org/abs/1608.08614
[75] U. Kacar and M. Kirci, "ScoreNet: Deep cascade score level fusion for unconstrained ear recognition," *IET Biometrics*, vol. 8, no. 2, pp. 109–120, Mar. 2019.
[76] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive Modeling*, vol. 5, no. 3, p. 1, 1988.
[77] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1139–1147.
[78] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

HAMMAM ALSHAZLY received the B.Sc. degree in computer science from South Valley University, Egypt, in 2006, the M.Sc. degree in computer science from the University of Mumbai, India, through a scholarship from the Indian Council for Cultural Relations (ICCR), in 2014, and the Ph.D. degree in computer science from South Valley University, in 2018. He is currently a Postdoctoral Researcher with the Institute for Neuro- and Bioinformatics, University of Lübeck, Germany. He is also working as an Assistant Professor with the Department of Mathematics, Faculty of Science, South Valley University. He has published articles in conferences and peer-reviewed journals, and works as a reviewer for several journals. His research interests include deep learning, biometrics, computer vision, machine learning, and artificial intelligence. He was awarded the Partnership and Ownership (ParOwn) Initiative in 2010 for a period of six months at Monash University, Australia. During his Ph.D. degree, he was awarded a Fulbright Scholarship for ten months to complete part of his research work at the University of Kansas, USA.

CHRISTOPH LINSE received the B.Sc. degree in physics from the Rheinische Friedrich-Wilhelms-Universität Bonn, Germany, in 2014, and the first M.Sc. degree in physics from the Norwegian University of Science and Technology, Norway, through a scholarship from Cusanuswerk, in 2016, and the second M.Sc. degree in computational science from the Institute for Neuro- and Bioinformatics, University of Lübeck, Germany, where he is currently pursuing the Ph.D. degree in computational neurosciences. He is also enrolled at the Zentrum für Künstliche Intelligenz Lübeck. Also, he works for the county Land Schleswig Holstein to qualify small and medium-sized businesses in Schleswig Holstein to digitalize their value added chain introducing state-of-the-art techniques from machine learning and deep learning. His research interests include computational neurosciences, machine learning, and deep learning.

ERHARDT BARTH (Member, IEEE) received the Ph.D. degree in electrical engineering from the Technical University of Munich, in 1994. He is currently a Professor of computer science and the Deputy Director of the Institute for Neuro- and Bioinformatics, University of Lübeck, Germany. He leads the research on human and machine vision at the Institute for Neuro- and Bioinformatics. He was a Research Associate with the Department of Communications Engineering, Munich, and a Visiting Fellow with the Department of Computer Science, Melbourne University, Australia, where he was supported by the Gottlieb-Daimler and Karl-Benz Foundation. He then was a Researcher at the Department of Medical Psychology, University of Munich, and a Klaus-Piltz Fellow at the Institute for Advanced Study in Berlin. In 1997 and 1998, he was a member of the NASA Vision Science and Technology Group, NASA Ames, Moffet Field, CA, USA. He has published more than 200 research articles in peer-reviewed journals and international conferences. His research interests include computer vision, machine learning, and artificial intelligence.

THOMAS MARTINETZ (Senior Member, IEEE) studied physics at the Technical University of Munich, Germany. He received the Ph.D. degree in theoretical biophysics from the Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, USA. From 1991 to 1996, he led the project Neural Networks for automation control at the Corporate Research Laboratories, Siemens AG, Munich. From 1996 to 1999, he was a Professor of neural computation with the Ruhr-University of Bochum and the Head of the Center for Neuroinformatics. He is currently a Professor of computer science and the Director of the Institute for Neuro- and Bioinformatics, University of Lübeck, Germany. He co-founded the software companies: Consideo, Pattern Recognition Company, and Gestigon. He has published more than 300 research articles in peer-reviewed journals and international Conferences. His research interests include neural networks, machine learning, artificial intelligence, and computational neuroscience.