A Mixture of Personalized Experts for Human Affect Estimation

Michael Feffer, Ognjen (Oggi) Rudovic, and Rosalind W. Picard

MIT Media Lab, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
{mfeffer,orudovic,roz}@mit.edu,

Abstract. We investigate the personalization of deep convolutional neural networks for facial expression analysis from still images. While prior work has focused on population-based ("one-size-fits-all") approaches, we formulate and construct personalized models via a mixture of experts and supervised domain adaptation approach, showing that it improves greatly upon non-personalized models. Our experiments demonstrate the ability of the model personalization to quickly and effectively adapt to limited amounts of target data. We also provide a novel training methodology and architecture for creating personalized machine learning models for more effective analysis of emotion state.

Keywords: mixture of experts, domain adaptation, personalized machine learning, residual networks

1 Introduction

In recent years, machine learning has become increasingly popular for performing analysis and generating predictions based on data in a variety of different areas, especially in healthcare and fields devoted to improving health and wellness [1, 2]. In the realm of affective computing, it has been applied to tasks such as automated analysis of persons' engagement, personality, and affective states during human-computer [3] and human-robot interaction [4]. For instance, as robots become increasingly complex and integrated into daily life, it will be important for them to perceive and understand not only human biometrics but also human emotional metrics. Enhanced perception of human emotions could enable robots to avoid actions that would worsen a human's emotional state and perhaps even influence them to act in a way that could improve a human's well-being. Moreover, in the advent of powerful machine learning capabilities for mobile devices, it is possible nowadays to perform emotion analysis through smartphone cameras. Therefore, a smartphone application could be programmed to detect a user's emotions and recommend strategies for dealing with negative emotions or actively attempt to improve the user's mood. Lastly, emotion analysis could be used for emotion and engagement detection and coaching for individuals with autism, who have inherent difficulties in reading others' emotions and expressing their own in a way that can be easily understood by neurotypicals.

Most machine learning approaches are successful because the models produced can generalize to unseen data and were trained on a plethora of existing data. However, most of the existing approaches ignore the fact that people express affect differently, even when they are part of the same culture. Therefore, learning a general predictor or classifier (the traditional "one-size-fits-all" approach) with data from one set of people typically underperforms when tested on people from a disjoint set but also on specific people within the training set. Moreover, it is difficult to obtain a large amount of training data for each target subject because providing labels for these data is costly in terms of time and resources [5]. Thus, improving machine learning approaches so that they can efficiently leverage small amounts of training data to adapt to each target subject is of large importance for increasing the model's effectiveness. To address these challenges, a number of works attempted model personalization [4]. The goal of model personalization is to leverage the individual-specific data in order to adapt a general classifier (the "one-size-fits-all" approach) learned from data of previously seen people (source subjects) to the profiles of specific individuals (target subjects). This has shown great improvements in a number of machine learning tasks related to human-data analysis (e.g., [6, 2, 1]).

In this paper, we have focused on a specific type of model personalization for estimation of facial affect (valence and arousal) using the notion of an ensemble of models and domain adaptation [7]. Specifically, we use the Mixture-of-Experts (MoEs) approach [8] to model the facial expression data (face images) of source subjects, for whom it is assumed that a large amount of image labels for the facial affect is readily available. We adopt MoEs where each expert represents one of the source subjects, which has greater modeling flexibility and improved ability to capture the large variation in facial expressions of different subjects compared to a single expert, which typically captures the average variation. While this approach performs better fitting of the source subjects, it is not guaranteed that this performance translates to previously unseen subjects (target), as confirmed in our experiments. To this end, we perform a supervised adaptation of the learned MoEs model using a varying portion of labeled data of all of the target subjects. We show: (i) that this approach achieves improved performance on the target subjects compared to a single expert model, and (ii) that it also outperforms the same model trained solely on the data of target subjects used to adapt the model. The latter is due to the ability of our approach to efficiently leverage the data of the source subjects. We demonstrate this in the context of deep neural networks that we tuned for "end-to-end" estimation of valence and arousal from still images of faces from the multimodal affect database REmote COLlaborative and Affective (RECOLA) database [9], used in the Audio/Visual Emotion Challenge and Workshop (AVEC) 2016 [10]. Note, however, that the focus of this work is not to outperform existing models in affect estimation but instead to examine how personalization and supervised domain adaptation can bolster current affect estimation models, an intersection of research areas that has yet to be explored.

In the sections that follow, we describe our approach to personalizing deep convolutional neural networks and a MoEs model. We first discuss related work both regarding the domain adaptation and MoEs. Then, we describe our mixture of personalized experts approach, followed by its experimental validation and derived conclusions.

2 Related Work

MoEs refers to a learning approach that involves training multiple "expert" subnetworks. These networks produce the same type of output for the same type of input, but they are trained on different subsets of data so that they are fine-tuned to specific contexts. At test time, all of them are given the same input, and an output for the overall network is created by combining the expert outputs in a certain way or somehow selecting the "best" output. Among the first proponents of this technique, [8] introduce a method of selecting an output by using a gating network that performs softmax activation to assign probabilities of randomly selecting the outputs of each of the expert networks for the current training example, after which an output is randomly chosen via a probability draw based on that distribution. Since then, it has been studied extensively for over twenty years, and in that time, expert models with other classifiers such as SVMs [11] and Gaussian Processes [12, 13], have also been researched. With regard to deep neural networks, a myriad of different network architectures have been explored, ranging from hierarchical and ensemble experts [14] to networks with infinite numbers of experts [15] and nested mixtures of experts that allow for deep learning [16]. Although our work is built upon the same framework of MoEs, it differs from existing approaches as we personalize the experts to each subject. Furthermore, we combine the learning of MoEs with supervised domain adaptation (DA) [7] in order to efficiently adapt the model to previously unseen subjects. While there is a large body of work on DA, we do not intend here to improve upon existing DA methods. We rather use its notion when adapting the target MoE model that we propose for the model personalization to target subjects. For a detailed review of existing DA approaches, see [7]. Also, even though model personalization has been researched in several previous contexts (e.g., self-reported pain analysis [6] and robot therapy for autism [4]), none of these methods have been explored using a mixture of experts architecture in the context of model personalization. To this end, we have adopted this approach in our personalized framework. We have taken particular inspiration from [16] to utilize a mixture of experts approach that outputs a weighted combination of expert outputs based on a gating network learned form source subjects, as will be elaborated in the following sections.

3 Methodology

3.1 Notation

We consider the following setting: we are given a number of training subjects (source), denoted as $P^{(s)} = \{p_1^{(s)}, \ldots, p_{n_s}^{(s)}\}$, where $id^{(s)} = 1, \ldots, n_s$ represents

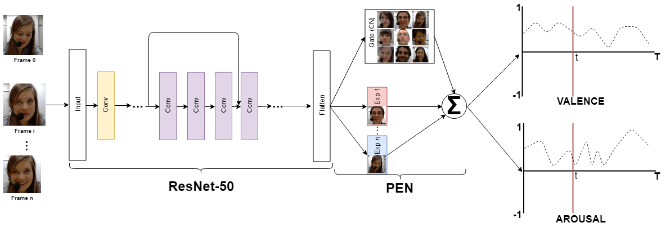

Fig. 1: The architecture of the proposed approach. The input is a subject's video and the outputs are his/her estimated valence and arousal levels. We first applied Faster R-CNN [17] to extract the face region from each raw image frame. The extracted faces were passed through a ResNet-50 [18], fine-tuned on source subjects' data. The obtained deep features were used as input to our personalized expert network (PEN) for automatic estimation of valence and arousal. This also contains a "gating network" (CN) that assigns different weights to each expert in the PEN during inference of new test images.

the subject id, and n_s is the number of the subjects. Data of these source subjects are used to learn a shared model optimized on these subjects. We are also given a number of test subjects (target) which were previously unseen by the shared model. These are denoted as: $P^{(t)} = \{p_1^{(t)}, \ldots, p_{n_t}^{(t)}\}$, and $id^{(t)} = 1, \ldots, n_t$. Then, starting from the shared model, the goal is to achieve the best possible performance on $P^{(t)}$. Furthermore, the data of each subject is stored as $p_i = \{X_i, Y_i\}$[1], where input features of the subject i are given by: $X_i = [x_{i1}; \ldots; x_{iN_i}] \in \mathcal{R}^{N_i \times D_x}$, where N_i is the number of available examples of the subject, and D_x is the input feature size. Note that these examples may be temporally correlated (in the case of video data) or be randomly sampled from independent observations of the subject. Likewise, the output labels (in our case, the levels of valence and arousal of the subject), are given as: $Y_i = [y_{i1}^v \, y_{i1}^a; \ldots; y_{iN_i}^v \, y_{iN_i}^a] \in \mathcal{R}^{N_i \times D_y}$, where $D_y = 2$. In what follows, we first describe how the shared model is learned from $P^{(s)}$, and used to estimate target affective states on $P^{(t)}$. Then, we propose an expert model, where each expert corresponds to one of the source subjects. The key to our approach is the model personalization step, where we propose a learning strategy designed to adapt the expert model to target subjects, using a varying portion of their (previously seen) data. Lastly, we provide details of the learning and inference in the personalized expert model.

3.2 Shared Model

We start by learning a shared model that is trained on data of all source subjects, without taking into account their id. This model is based on a deep architecture composed of the layers of a residual network (ResNet) [18], a pre-trained

[1] For notational simplicity, we drop here the dependence on the source/target subjects

deep network commonly used to extract the most informative features for object classification [18]. We used the ResNet-50 architecture composed of multiple three-layer "bottleneck" building blocks (containing 50 layers in total) described in the original ResNet paper [18]. When beginning training, we initialized the layer weights corresponding to weights that yield published optimal performance on the ImageNet dataset [19]. However, we use all of the layers of the network but the last (i.e. the softmax layer) as it was optimized for classification of various object categories such as "laptop" and "orange". We instead replace the last layer with an data-uninformed fully-connected dense layer with linear activation, which we use to fine-tune the ResNet weights for the target task: the estimation of valence and arousal from face images (see Fig. 1). This architecture receives as input the face images of source subjects (x) and passes the most discriminative (deep) facial features (z) to the regression layer in the output through the following mapping: $x \rightarrow z \rightarrow \tilde{y}$, where $\tilde{y}$ are the estimated levels of valence and arousal. The training of the shared model is divided into two stages. First, the whole network (ResNet included) is optimized for estimation of y. Then, we freeze all of the (fine-tuned) layers of the ResNet ($W^{r-net} \in \mathcal{R}^{D_x \times D_z}$) and additionally fine-tune the last (regression) layer ($W^s \in \mathcal{R}^{D_z \times D_y}$). We experimented with different learning strategies and found that this one performed the best. This is because of the large number of parameters that need to be tuned simultaneously. Due to this, the network underfits the last layer, so we overcome this by additional tuning of the last layer. The resulting network, referred to as the shared network (SN), is used to initialize the expert network (EN), which is then adapted to the population of target subjects as described below.

3.3 Personalized Expert Network (PEN)

The learning of the expert network is accomplished using the Mixture-of-Experts (MoEs) approach, where an expert network (EN) is comprised of a set of layers called "experts" that are denoted as $e_1, \ldots, e_n$. Furthermore, an EN is also comprised of a "gating network" denoted as CN (which in our case is a person selector network). Its output is used to weight the contribution (relevance) of each expert during the inference stage. In our personalized model setting, during the model training on source subjects, each expert corresponds to one training subject (thus, n_s experts). This personalization yields a personalized expert network (PEN). Each expert is modeled using a feed-forward network with fully connected linear activations, as used for the SN, but each with their own parameters. Thus, the following mapping is learned: $x \rightarrow z \rightarrow \tilde{y}^e = [\tilde{y}^e_1 \; \ldots \; \tilde{y}^e_{n^{(s)}}]$, where $\tilde{y}^e_i$ are the valence/arousal estimates by the i-th expert. Likewise, the CN aims to learn the mapping: $x \rightarrow z \rightarrow h \rightarrow \tilde{c} = [\tilde{c}_1 \; \ldots \; \tilde{c}_{n^{(s)}}]$, where $\tilde{c}_i$ is the (normalized) weight for the i-th expert during the model learning. More specifically, the CN is designed as a two-layer network. The first layer is a fully-connected feed-forward linear activation network. The linear activations are then passed through a softmax layer, providing the probability that the input sample comes from one of n_s source subjects and thus assuring that the outputs of the CN sum to one. More formally, given an input x to the ResNet, the output of the

PEN is defined as:

$$\tilde{y} = \sum_{i=1}^{n_s} \tilde{c}_i \cdot \tilde{y}_i^e = \tilde{c} \otimes \tilde{y}^e, \tag{1}$$

where $\tilde{y}$ is the weighted combination of the individual experts. The output of the CN, given the activations (z) of the ResNet is obtained as:

$$\tilde{c} = \text{softmax}(h) \text{ and } h = \text{fcl}(z; W^s), \tag{2}$$

where z is passed through the fully connected layer (fcl), the output of which is fed into the softmax function to yield a categorical probability distribution over the source subjects. Note that during training of the PEN, the targets for the CN output are the subjects' ids encoded via the 1-hot encoding (e.g., $c = [0, 1, 0, \ldots, 0]$ for subject $i = 2$). Similarly, each expert $i = 1, \ldots, n^{(s)}$ produces estimates for target affective dimensions as:

$$\tilde{y}_i^e = \text{fcl}(z; W_i^e), \tag{3}$$

where z is multiplied by a trainable weight matrix W_i^e for expert i.

The network personalization is attained by using the prior knowledge about the source subjects: each expert is supposed to represent one of the subjects, and CN performs the selection of that expert during the model learning. Therefore, given the training data of $n^{(s)}$ source subjects, the overall loss is the sum of losses due to differences between y and $\tilde{y}$ (the weighted combination of expert outputs) as well as losses from the CN. However, in practice this loss does not enforce the sparsity on the selector's weights $(\tilde{c})$, which may result in the learned PEN expending too much modeling power of each expert on trying to fit the data of all source subjects. In turn, we may end up with an expert that is unable to specialize in individual characteristics of the subject, resulting in an average model that is suboptimal. To overcome this, we introduce the L-1 sparsity constraint on the output of the CN, but this cannot be done directly because the outputs of that layer always sum to one. Instead, we enforce the sparsity on the activations of the fcl of the SN. This leads to the following joint loss being optimized during the parameter learning:

$$\alpha = \alpha^y + \lambda_0 \alpha_c^s + \lambda_1 \alpha_r^s, \tag{4}$$

where α^y is the mean-squared error (MSE) loss between the PEN estimates and the ground-truth labels for valence and arousal (y). (λ_0, λ_1) are regularization parameters that are optimized on the validation data. They control the trade-off between the model performance and the penalty terms: α_c^s, which ensures that each expert focuses on its corresponding subject, and α_r^s further ensures this through the sparsity constraint. These individual losses are defined as:

$$\alpha^y = \frac{1}{N^{(s)}} \sum_{i=1}^{n^{(s)}} \sum_{j=1}^{N_i^{(s)}} (y_j^i - \tilde{y}_j^i)(y_j^i - \tilde{y}_j^i)^T, \tag{5}$$

and $N_i^{(s)}$ and $N^{(s)}$ are the number of training data per source subject and overall, respectively. The selector loss is given by:

$$\alpha_c^s = \frac{1}{N^{(s)}} \sum_{i=1}^{n^{(s)}} \sum_{j=1}^{N_i^{(s)}} H(c_j^i, \tilde{c}_j^i), \tag{6}$$

where $H(\cdot,\cdot)$ is the cross-entropy loss that is commonly used for discrete variables, as is the case here. Finally, the standard L-1 sparsity is enforced via:

$$\alpha_r^s = \frac{1}{N^{(s)}} \sum_{i=1}^{n^{(s)}} \sum_{j=1}^{N_i^{(s)}} |h_j^i|_{L_1}. \tag{7}$$

Note that this loss treats each activation/image frame independently, and therefore, no structure (prior information) about the source subjects is directly modeled. Nevertheless, this should still result in the learned activations being sparse, on average, for different face images of the subjects.

3.4 PEN: Supervised Adaptation to Target Population

Once the PEN parameters are optimized for the source subjects, our goal is to achieve the best performance on target subjects that the PEN has not seen before. In the traditional supervised machine learning approach, this would be evaluated using the network learned solely from the data of the source subjects. However, this typically leads to the learned model attaining a lower performance on target subjects than on the source subjects, as expected. Also, since the PEN is "tuned" to the latter, it may even more easily overfit those subjects, leading to the comparable to or worse performance than that of the SN trained on the target subjects. This is despite the modeling flexibility introduced by the local (person) experts, which allows the PEN to better fit the source subjects. Conversely, in the proposed personalized learning approach, we adopt the supervised domain adaptation approach [20], where the target subjects are treated as another domain assumed to be different from the one used to train the PEN. To investigate this, we pool the data of target subjects and use a varying portion of this data (with approximately equal amounts of data from each subject) to "tune" the PEN to those subjects. This is driven by two assumptions. First, we should be able to adapt the PEN to target subjects using a (significantly) smaller number of target data than originally used from the source. This is mainly because the network has been pre-trained on the latter and should be able to adapt to new subjects more easily. Second, while the same assumption may hold for the SN, the proposed PEN is more flexible in its modeling power (due to the multiple experts). This in turn should allow it to better adapt to the previously unseen subjects by focusing on their individual characteristics, thus, avoiding the limitations of the "one-size-fits-all" approach. Formally, this is attempted by fine-tuning the PEN to the population of target subjects via the following adaptation loss:

$$\alpha_{ad} = \alpha^{y^t} + \lambda_1 \alpha_r^{s,t}, \tag{8}$$

This supervised adaptation loss uses a small number of labeled data of target subjects (x^t, y^t) to adjust the PEN parameters. It is important to note that we do not use the ids of target subjects during the model adaptation, thus, the parameters of the CN are optimized in an unsupervised fashion by setting $\alpha_c^s = 0$. However, we still impose the sparsity constraint on the CN parameters. We also do not further optimize the ResNet parameters - these are rather used as the feature extractor during the adaptation stage. The benefits of this are two-fold. First, we preserve the privacy of the target subjects[2]. This is important in the context of many applications where the user does not want to reveal his/her identity, while still being able to receive estimates of target affective states. Second, instead of completely "overwriting" the previously learned PEN, the adapted model performs the actual adaptation of the model rather than learning it from scratch. This is motivated by the assumption that the PEN model has seen a large amount of labeled data from source subjects, thus encapsulating valuable knowledge about the affect expressions of different subjects. In this way, more robust adaptation of the PEN to new subjects is expected. Ideally, when $n^{(s)} > n^{(t)}$, PEN should be able to specialize one expert to each target subject, while compensating the lack of target subject data with the knowledge extracted from the source subjects.

3.5 Learning, Inference and Implementation Details

We first summarize the learning and inference in the proposed models, followed by the implementation details of our deep architecture. We start with the learning of the SN. The learning of the model parameters $\{W^{r-net}, W^s\}$ is performed in two steps: (i) The joint fine-tuning of the ResNet with the parameter optimization in the appended fcl, used for estimation of valence and arousal. (ii) The additional fine-tuning of the fcl parameters $\{W^s\}$, while freezing the ResNet parameters i.e., $\partial W^{r-net} = 0$. For (i) W^{r-net} was initialized to weights corresponding to optimal training on the ImageNet dataset, and W^s was initialized randomly. Our implementation appended the SN to the last flattening layer of the ResNet and trained all of these layers together. For (ii) we took our best model from (i) (based on validation set performance) and froze every layer in the architecture besides the SN. After more training, we saved this fully-adapted architecture and used it as the starting point for expert layers going forward.

To learn the PEN parameters, we first initialized each expert using the weights of the SN as $W^{e_i} \leftarrow W^s$, $i = 1, \ldots, n^s$. This ensures that individual experts do not overfit the corresponding source subjects, which could adversely affect their generalization to target subjects (we describe this below). The initial learning of the CN was done in isolation from the rest of the network. Only the outputs of the fine-tuned ResNet were used as input (z), and the $W^{c,0}$ was optimized by minimizing the loss $\alpha_c^s + \lambda_1 \alpha_r^s$ using the 1-hot encoding of the source

[2] For instance, only the ResNet features of target subjects need be provided as input to the adapted model, as original face images cannot be reconstructed from those features.

Algorithm 1 Personalized Experts Network (PEN)

```
Source Learning Input: Source persons data $P^{(s)} = \{p_1^{(s)}, \dots, p_{n_s}^{(s)}\}$
step 1: Fine-tune ResNet weights ($W^{r-net}$) and optimize SN ($W^s$)
step 2: Freeze ResNet weights and fine-tune SN ($W^s$)
step 3: Initialize the experts ($W_i^e \leftarrow W^s$) and optimize PEN ($W^s, W^c$)
Target Adaptation Input: Target persons data $P_{ad}^{(t)} \leftarrow n\%$ of $P^{(t)} = \{p_1^{(t)}, \dots, p_{n_t}^{(t)}\}$
step 1: Fine-tune PEN weights ($W^s, W^c$) using adaptation data $P_{ad}^{(t)}$

Inference Input: Unseen target persons data $P_{un}^{(t)} = P^{(t)} \cap P_{ad}^{(t)}$
Output: $(\tilde{y}^v, \tilde{y}^a) \leftarrow \mathrm{PEN}(P_{un}^{(t)})$
```

subjects' ids as ground-truth labels (c). Then, the joint learning of the PEN was performed by minimizing the loss in Eq.(1). We noticed that the model with the individually tuned experts to data of each subject (thus, in isolation from the selector) generalized worse to target subjects than when only the joint learning of the selector and the experts was attempted. For this reason, we report our results only for the latter setting. Also, we noticed that doing the joint learning for a large number of epochs even led to overfitting of source subjects (on their left-out portion of the data). For this reason, we used the early stopping strategy, which prevented the model from overfitting after only five epochs. This model was subsequently used for further adaptation to target subjects by minimizing the PEN loss (Eq.(8)) on the adaptation data of target subjects – see Sec. 4. These learning and inference steps are summarized in Alg. 1.

We implemented the PEN architecture using the Keras API [21] with the Tensorflow [22] back-end. For the soft-max and fcls, we used the existing implementations. The layer sizes were 2048 x 9 (for a total of 18441 parameters including the 9 offsets) for the CN and 2048 x 2 (for a total of 4098 parameters including the 2 offsets) for a given expert. The reweighting part of the weighted sum was performed via a custom Lambda layer that took the tensors output by the CN and the experts as input and scaled the outputs of each expert by the corresponding CN output. Afterwards, the scaled components were summed via an Addition layer and output as the overall network output. During training with source data, mean-squared error was used to train with this overall output, and categorical cross-entropy loss was used to train with the output of the CN. However, to implement the PEN loss, we created a custom loss function that performed L-1 regularization on the pre-softmax- activation CN output by taking the mean of the absolute value of the pre-activation tensor and summing it with the categorical cross-entropy loss. The parameter optimization was then performed using the standard back-propagation algorithm and Adadelta optimizer with the default parameters. The details of the employed validation settings are provided in the description of the experiments.

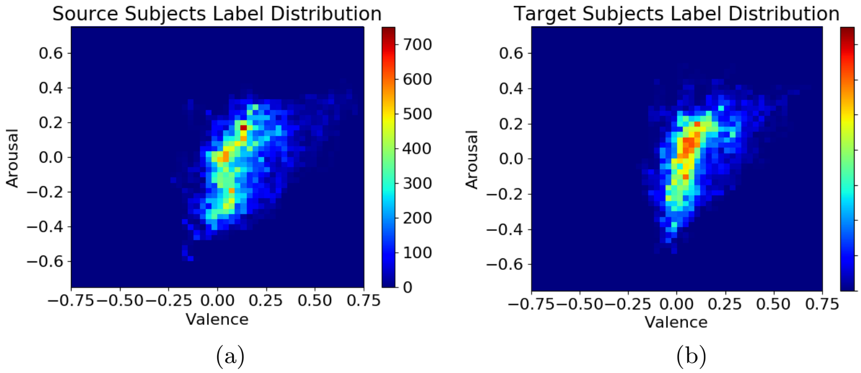

Fig. 2: The joint distribution of the labels of affective dimensions: valence and arousal, in source and target subjects. By personalizing the PEN using adaptation data of target subjects, we reduce the difference between two distributions.

4 Experiments

To demonstrate the effects of the model personalization using the proposed PEN approach, we used image sequences from the multimodal affect database REmote COLlaborative and Affective (RECOLA) database [9] , used in the Audio/Visual Emotion Challenge and Workshop (AVEC) 2016 [10] This database contains four modalities or sensor-signals: audio, video, electro-dermal activity (EDA), and electro-cardiogram (ECG). The data are synchronized with the video modality and coded by five human experts. Specifically, the gold standard labels (i.e., the aligned codings) for two affective dimensions - valence and arousal - are provided by the database creators. The time-continuous codings for each dimension are provided on a scale from -1 to +1. For our experiments, we used the publicly available data partitions from AVEC 2016, namely, training (9 subjects) and development (9 subjects) sets. We refer to these as source and target persons. The video of each person is 5 mins long (25fps), resulting in ∼7k image frames per person after the face detection using Faster R-CNN [17]. We used the processed face images as input to the models, and the output was the estimated levels of valence and arousal per frame. The models' performance was evaluated in terms of Root-mean-square-error (RMSE) and concordance correlation coefficient (CCC), both of which were used as competition measures in AVEC and were computed on the pairs of model estimates and the gold-standard labels.

We performed the following experiments: (i) the SN and PEN models trained and tested on the source subjects ($P^{(s)}$), in order to evaluate the modeling power of the latter when fitting the data. We denote these models as s-SN and s-PEN. (ii) These models were then adapted using the adaptation data ($P_{ad}^{(t)}$), a varying portion of the data from target subjects, and evaluated on the non-overlapping data of the target subjects. The data of target subjects were split into the adaptation and test data at random. Specifically, we formed $P_{ad}^{(t)}$ by incrementally sampling the following amount of data: $n = 5\%, 10\%, 20\%, 30\%$ and 50% from each target subject and then combining the data across the subjects. For example, from 7k frames per target subject, $n = 5\%$ led to $P_{ad}^{(t)}$ of size 350*9=3150 images, and for $n = 50\%$ the $P_{ad}^{(t)}$ size was ten times larger.

For testing, we always used the same (nonoverlapping) 50% of target subject data. The goal of these comparisons was to assess the models' behavior when using a different amount of target subject data to adapt the deep networks. The main premise here is that due to the modeling flexibility of the PEN, it would be able to better adapt to the target subjects than would SN. To show the benefits of using the data of source subjects to learn the (non-adapted) models, we also tested the SN and PEN model architectures trained from scratch on the 5 different adaptation sets $P_{ad}^{(t)}$ formed from the $n\%$ of target data as described above and evaluated on the test set, i.e. the left-out 50% of target subject's data. (iii). We refer to these settings as target SN (t-SN) and target PEN (t-PEN), respectively.

To form the base models, we first trained the s-SN together with the ResNet (see Alg.1). This was accomplished using 80% of the source data (evenly sampled from each subject) while the remaining 20% were left out for the model validation. We found that 10 epochs were sufficient to fine-tune the ResNet without overfitting it due to its large number of parameters and the limited number of data used to tune the network. Further optimization of the SN and PEN configurations was performed using 30 epochs for training the models, which was enough for the models' loss to converge. To select the regularization parameters when training s-PEN, we cross-validated (λ_0, λ_1) using the following values $\{10^{-4}, 10^{-3}, .., 0, 1, 10, 100\}$, with 10^{-3} performing the best for both. During model adaptation and training of the target models, we used the same regularization parameters. Note that for these models, no subject id was provided during the adaptation/training, as we assumed that these are available only for the source subjects.

Table 1 compares our networks initially trained with source data, s-SN and s-PEN, with the t-SN and t-PEN, trained from scratch using only $n\%$ of target data, as mentioned above[3]. The results show that the initial training of the SN and PEN on the source data improved performance on target test data, compared to the t-SN and t-PEN. This evidences that the proposed models were able to efficiently leverage the data of the source subjects during the estimation. Moreover, the s-PEN approach eventually outperforms the SN architecture after as little as 10% of target (adaptation) data, due to its more flexible architecture that allowed it to easier adapt to target population data. Furthermore, we observe that with even as few as 5% of target subjects' data, both models' performance improves largely, with the s-PEN improving more advantage as more adaptation data become available. We assume that these (supervised) adaptation data are sufficient to constrain the feature/label space of the source models,

[3] Note, however, that the Resnet used to extract the features for these models was fine-tuned using the labeled source data.

$n[\%]$		0	5	10	20	30	50
Valence	s-SN	**0.72**	**0.80**	0.80	0.82	0.82	0.82
	s-PEN	0.71	**0.80**	**0.82**	**0.84**	**0.85**	**0.86**
	t-SN	N/A	0.71	0.77	0.8	0.81	0.82
	t-PEN	N/A	0.75	0.79	0.81	0.82	0.83
Arousal	s-SN	**0.66**	**0.79**	0.80	0.81	0.82	0.82
	s-PEN	0.65	**0.79**	**0.81**	**0.83**	**0.84**	**0.85**
	t-SN	N/A	0.73	0.77	0.79	0.81	0.81
	t-PEN	N/A	0.76	0.79	0.8	0.81	0.82

Table 1: Performance on target test data in terms of CCC after adapting the networks with $n\%$ of (non-overlapping) target data.

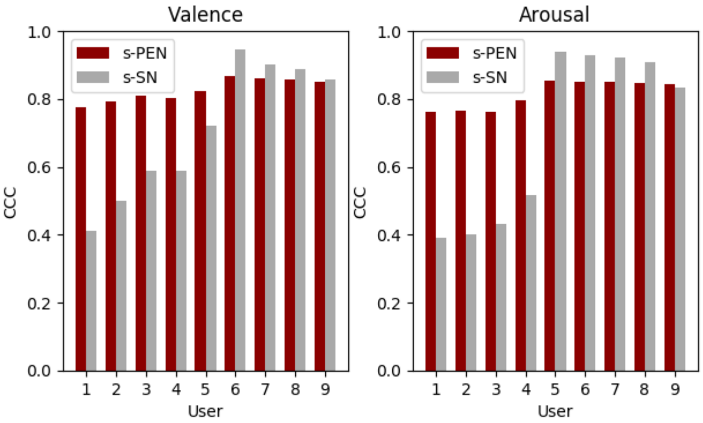

Fig. 3: Per-subject valence and arousal estimation performance on target test data of source-trained models adapted with limited target data. The s-PEN model has more consistent performance than the s-SN model over all of the target subjects.

rendering more efficient models for the target population. We also note the high performance of the t-SN and t-PEN, even with only 5% of the target data. We attribute this to the fact that they use the same ResNet feature extractor that was fine-tuned (via the s-SN) with the source labeled data. At the same time, we also note that s-SN and s-PEN have been trained on significantly more data. This, in turn, allowed the s-PEN to outperform the t-PEN by larger margin than is the case with their SN versions.

To analyze the effects of the model personalization to the target population, in Fig. 3 we show the CCC values of the s-SN and s-PEN models per target subject. We averaged the per-subject CCC values for both valence and arousal across all of the adaptation data sizes, and sorted the subjects based on the absolute difference in the models' performance for each subject. We observe

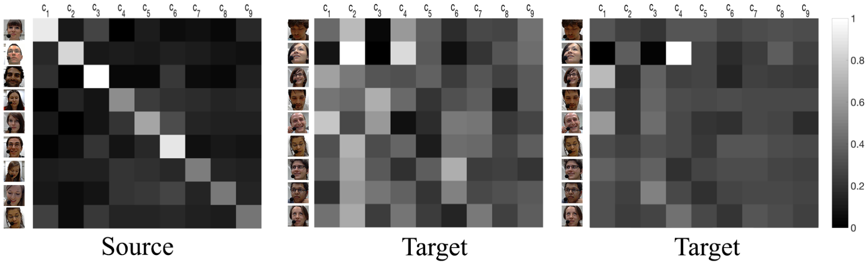

Fig. 4: Sparse Combinations of Experts via CN. Left: the selector learns the weighting of the outputs for the source subjects during source training. Center: the selector weights are effectively random yet sparse for the target subjects from source training. Right: after some fine-tuning on target, the selector begins to lose its sparse weighting despite regularization.

that on 4/9 subjects, s-PEN largely outperforms s-SN in estimation of both valence and arousal levels. On the remaining subjects, s-SN outperforms t-PEN - however, these differences are less pronounced. For instance, the s-PEN model consistently produces average CCC values for each subject that are around 0.8. This experiment shows limitations of the "one-size-fits-all" machine learning architecture on certain subjects. By contrast, the modeling flexibility of the s-PEN allows it to better fit the data distribution of the target population.

We depict the effects of our custom loss function based on the the L-1 (sparsity) regularization as well as how the CN was learning a combination of the source subjects in Fig. 4. These three plots show the progression of the average outputs of the CN weights per subject as our algorithm advances. As seen in the first image, the source weights are nearly an identity matrix after training only on the source subjects. This allowed the s-PEN to specialize each expert to one source subject, while also sharing the knowledge. However, when evaluated on target subjects, the selector produces different weights after unsupervised fine-tuning to those subjects (i.e. it is ignorant of the subject ids). Compared to the third image, these "subject" weights are still more sparse than when the training is done using data of target subjects. This is because the former uses the pre-trained selector network, resulting in the more sparse weights. By contrast, without leveraging this information, the s-PEN finds it more challenging to specialize in target subjects (as measured by the distribution of its CN weights ($c_1 \ldots c_9$)) after the fine-tuning, despite the regularization of its weights (perhaps, due to the lack of subject ids). On the other hand, we found that by increasing the level of regularization adversely affects the model's performance, diminishing the role of the valence-arousal estimation loss.

Finally, in Table 2, we have included the s-PEN model's performance alongside the performance reported recently by [23], where a ResNet-50 with Gated Recurrent Unit (GRU) networks for sequence estimation is used for estimation of valence and arousal from videos of the same target subjects. We show that

Model	Valence	Arousal	Avg.
s-PEN (0% of fine-tuning data)	0.71	0.65	0.68
s-PEN (5% of fine-tuning data)	0.80	0.79	0.80
End2You	0.58	0.41	0.50
AVEC 2016 Baseline	0.61	0.38	0.50

Table 2: Comparison to End2You and AVEC 2016 Baseline [23].

the proposed s-PEN exceeds their reported performance by large margin. However, these results may not directly be comparable because of possibly different evaluation settings.

5 Conclusions

In summary, we propose a novel strategy for the personalization of deep convolutional neural networks for the purpose of valence and arousal estimation from face images. The key to our approach is the personalization of the mixture of experts architecture using a limited amount of data of target subjects. These personalized models have clear advantages over the compared single-expert ("one-size-fits-all") models in terms of how well are able to adapt to the target population when using limited amounts of labeled data from target subjects. Given the limitations involved in obtaining annotated valence and arousal data due to cost of expert labor and large variations in levels of expressiveness between people, model personalization can be key in working with limited data with many different domains. The audio-visual data we used come from sessions limited to 5 minutes, yielding $\sim$7k image frames per subject, and we randomly sampled and split this data into non-overalapping training, adaptation, and test sets. However, ideally the system would have access to multiple sessions, allowing the proposed model to actively personalize as the interactions progress, and give us enough sessions so that we could draw samples that are further apart in time and less likely to be correlated. While minimizing correlation is not as critical in this work as in non-personalized situations, future work should explore how different methods of pseudo-random sampling of frames for constructing the adaptation and the hold-out test sets affect the results.

Acknowledgments

The work of O. Rudovic has been funded by the European Union H2020, Marie Curie Action - Individual Fellowship no. 701236 (EngageMe).

References

1. Peterson, K., Rudovic, O., Guerrero, R., Picard, R.W.: Personalized gaussian processes for future prediction of alzheimer's disease progression. NIPS Workshop on Machine Learning for Healthcare (2017)

2. Jaques, N., Rudovic, O., Taylor, S., Sano, A., Picard, R.: Predicting tomorrow's mood, health, and stress level using personalized multitask learning and domain adaptation. In: IJCAI Workshop. (2017)
3. Zeng, Z., Pantic, M., Roisman, G.I., Huang, T.S.: A survey of affect recognition methods: Audio, visual, and spontaneous expressions. IEEE TPAMI (2009)
4. Rudovic, O., Lee, J., Dai, M., Schuller, B., Picard, R.: Personalized machine learning for robot perception of affect and engagement in autism therapy. arXiv preprint arXiv:1802.01186 (2018)
5. Mollahosseini, A., Hasani, B., Mahoor, M.H.: Affectnet: A database for facial expression, valence, and arousal computing in the wild. IEEE TAC (2017)
6. Martinez, D.L., Rudovic, O., Picard, R.: Personalized automatic estimation of self-reported pain intensity from facial expressions. In: IEEE CVPR'W. (2017)
7. Csurka, G.: Domain adaptation for visual applications: A comprehensive survey. Advances in Computer Vision and Pattern Recognition (2017)
8. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural computation **3**(1) (1991)
9. Ringeval, F., Sonderegger, A., Sauer, J., Lalanne, D.: Introducing the recola multimodal corpus of remote collaborative and affective interactions. In: IEEE FG (Workshops). (2013)
10. Valstar, M., Gratch, J., Schuller, B., Ringeval, F., Lalanne, D., Torres Torres, M., Scherer, S., Stratou, G., Cowie, R., Pantic, M.: Avec 2016: Depression, mood, and emotion recognition workshop and challenge. In: Proc. of the 6th International Workshop on Audio/Visual Emotion Challenge, ACM (2016)
11. Collobert, R., Bengio, S., Bengio, Y.: A parallel mixture of svms for very large scale problems. In: NIPS. (2002)
12. Shahbaba, B., Neal, R.: Nonlinear models using dirichlet process mixtures. JMLR (2009)
13. Theis, L., Bethge, M.: Generative image modeling using spatial lstms. In: NIPS. (2015)
14. Yao, B., Walther, D., Beck, D., Fei-Fei, L.: Hierarchical mixture of classification experts uncovers interactions between brain regions. In: NIPS. (2009)
15. Rasmussen, C.E., Ghahramani, Z.: Infinite mixtures of gaussian process experts. In: NIPS. (2002)
16. Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In: ICLR. (2017)
17. Jiang, H., Learned-Miller, E.: Face detection with the faster r-cnn. In: IEEE FG. (2017)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE CVPR. (2016)
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012) 1097–1105
20. Jiang, J.: A literature survey on domain adaptation of statistical classifiers. URL: http://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey **3** (2008)
21. Chollet, F., et al.: Keras (2015)
22. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: OSDI. (2016)
23. Tzirakis, P., Zafeiriou, S., Schuller, B.W.: End2you–the imperial toolkit for multimodal profiling by end-to-end learning. arXiv preprint arXiv:1802.01115 (2018)