Semi-supervised Emotion Recognition using Inconsistently Annotated Data

S Happy, Antitza Dantcheva, Francois Bremond

Semi-supervised Emotion Recognition using Inconsistently Annotated Data

S L Happy, Antitza Dantcheva, and Francois Bremond
Inria, Université Côte d'Azur, France
{s-l.happy, antitza.dantcheva, francois.bremond}@inria.fr

***Abstract*— Expression recognition remains challenging, predominantly due to (a) lack of sufficient data, (b) subtle emotion intensity, (c) subjective and inconsistent annotation, as well as due to (d) *in-the-wild* data containing variations in pose, intensity, and occlusion. To address such challenges in a unified framework, we propose a self-training based semi-supervised convolutional neural network (CNN) framework, which directly addresses the problem of (a) limited data by leveraging information from unannotated samples. Our method uses 'successive label smoothing' to adapt to the subtle expressions and improve the model performance for (b) low-intensity expression samples. Further, we address (c) inconsistent annotations by assigning sample weights during loss computation, thereby ignoring the effect of incorrect ground-truth. We observe significant performance improvement in *in-the-wild* datasets by leveraging the information from the *in-the-lab* datasets, related to challenge (d). Associated to that, experiments on four publicly available datasets demonstrate large performance gains in cross-database performance, as well as show that the proposed method achieves to learn different expression intensities, even when trained with categorical samples.**

I. Introduction

Facial expression recognition aims at inferring emotions based on visual cues from face images. In spite of recent advancements in this field, following set of challenges remain open. (a) *Lack of* sufficient *annotated data* is a major limitation in emotion recognition. At the same time, we have that manual annotation is subjective (e.g., intra and inter-individual variation), as well as time-consuming. (b) Often large affective datasets are created by crowdsourcing [1], which results in *noisy and inconsistent annotation* due to inexperienced annotators. Utilizing wrongly labeled samples often negatively affects the model performance. Contrarily, (c) the *in-the-lab* datasets are created under *constraint environments* with proper annotation by experts. However, such datasets predominantly contain only *peak expression images* posed by actors, which is far from realistic. Consequently, a model trained with such data often fails when applied to real-world applications mainly due to low sample size and lack of expression intensity variation in the training data. Further, unlike 'object classification', multiple emotions may coexist in real-life data, raising false alarms in blended or mixed emotions [2], [3]. Given that emotion classes are not mutually exclusive but collectively exhaustive, it is important for the model to learn accurate emotion intensities for all emotion-classes. Finally, (d) in contrast to *in-the-lab* data, real-world data (*in-the-wild*) contains images with different illumination, facial pose, occlusion, stemming from various sensor-types, and other undesired factors, such as low resolution and blurred images.

This work was funded by the French Government (National Research Agency, ANR), under Grant ANR-17-CE39-0002.

Over the past years, convolutional neural networks (CNNs) have achieved outstanding performances in many domains of application, including facial expression recognition [4], [5], [6], [7], due to their powerful capacity for modeling complex input patterns. *Transfer learning* addresses the limited data problem in emotion recognition [6], [8] by using a pretrained network trained with large face data in the context of face recognition. Some works exploit the ample unlabeled face data along with the labeled data in a weakly or semi-supervised approach. CNN models based on *semi-supervised learning* (SSL) [9], [10] have been explored, demonstrating the effectiveness of emotion classification frameworks by leveraging information from unlabeled data. In *self-training* [11], an initial model selects the high confidence samples from unlabeled data, which is further utilized to update the model. In this work, we address the limited data problem using both transfer learning and self-training, thereby maximally exploiting the information in unlabeled data.

Training with one-hot vectors forces CNNs to predict one of the classes with high confidence. The presence of noisy or inadequate data overfits the model with one-hot encoding. Label smoothing [12] renders models less confident about their associated predictions and acts as a natural regularizer. We here use *successive label smoothing*, in order to allow the model to automatically adapt to different emotion intensities. In other words, we use label smoothing on the model predictions for the labeled data and utilize them for further parameter update. Thus, our model output is consistent with the emotion intensity, by adapting to subtle and mixed emotion intensities.

Towards adapting the model to noisy and inconsistent annotated samples, the literature suggests eliminating hard samples [13], [14] or assigning weights to samples [15] for further parameter tuning. In our SSL approach, we assume that the labeled data contains noisy annotations, whereas the selected unlabeled data used in training process is clean. Thus, we assign weight to each labeled sample based on its training loss, which indicates the *sample importance*. Our model allows the sample weight values to grow or shrink dynamically based on model parameters at every iteration. By decreasing the weight of hard examples, the model becomes robust to inconsistent and noisy labeled data during model training.

We firstly train an initial model to achieve sufficient accuracy in predicting emotion classes. This model is further employed for the purpose of classifying unlabeled data and including a fraction of high confidence samples in the

training process in the subsequent iteration. This allows us to leverage information from ample unlabeled data towards addressing the low sample size problem. The combination of successive label smoothing and the use of predicted class probability of unlabeled data as the true distribution, enables the model to automatically adapt to different emotion intensities. Thus, prediction scores from our model resemble the emotion class probabilities, allowing the recognition of low intensity and blended emotions. Further, we assign sample weights based on the training loss and the model is updated considering the importance of each sample. Thus, noisy and inconsistent labeled data is ignored in the training process, due to their low sample importance. The combined effect of SSL, successive label smoothing, and sample importance assignment learns a robust model, which inherently tackles the issues raised by the *in-the-wild* data.

Contribution In summary, our main contributions are the following.

- We propose a self-training based SSL approach to improve emotion recognition performance by leveraging information from the unlabeled data. As opposed to using the predicted class of unlabeled samples as ground-truth [16], we use the predicted class distribution as the true distribution in our model.
- We use successive label smoothing to adapt to expression images with different intensities. We demonstrate that our model adapts to expression intensity from the data, annotated with discrete expression categories.
- We incorporate insights from sample weighting [15], [14], in order to make our model robust against noisy and inconsistent annotation. Contrarily to former approaches [15], [14], our model initially assigns equal weight to each sample, which can further increase or decrease based on the training and validation loss.
- We report performances, outperforming state-of-the-art results on four datasets. We show that the performance of *in-the-wild* datasets improve largely from using the *in-the-lab* datasets as unlabeled data. The proposed method also achieves a significant improvement in cross-database experiments.

II. Related Work

A. Deep Learning-based Facial Expression Recognition

Given the powerful ability to model complex class representation, CNNs have outperformed and replaced hand-crafted feature extraction methods in many applications, including facial expression recognition [5], [17], [18], [19]. Most CNN methods attempted to resolve emotion recognition challenges inspired by methods, proposed in early literature. For example, CNN based models have been proposed for improving emotion recognition by encoding temporal appearance and geometry features [4], by jointly learning feature representation-selection-classification [17], by embedding facial representation of both peak – non-peak intensity samples [5], by jointly learning expression and identity related features [7], etc.

To address the limited data-size problem, Oquab et al. [20] proposed the transfer of mid-level image representation for related source and target domains. The network parameters learned on large-scale data generalizes the input space and transferring such parameters could significantly improve the performance of a task with a limited amount of training data. To improve emotion recognition performance with limited data, Ng et al. [8] used the network weights trained on ImageNet. Similarly, FaceNet2ExpNet [6] fine-tuned FaceNet in order to capture high level expression semantics. In our framework, VGG-Face model [21] is used to achieve knowledge transfer. VGG-Face uses a VGG-16 architecture and was trained with 2.6 million face images for face recognition applications and is a popular network choice for face related applications [6].

B. Semi-Supervised Learning (SSL)

SSL methods use both labeled and unlabeled data to overcome the limited data problem. Coates et al. [22] provided an intuition that data distribution from unlabeled data can be leveraged towards improving model performance. The performance of stacked auto-encoders was improved by initially training with unlabeled data followed by fine-tuning the model using labeled data [23]. Moreover, the use of adversarial networks, which learns from abundant unlabeled data [24], has been very well accepted.

Self-training or incremental SSL method [11], [16] is popular in deep learning classification tasks where the unlabeled data augment the limited annotated data. Self-training starts with optimizing model parameters with labeled data followed by classifying and selecting unlabeled data with high confidence scores for further training of the model. Co-training is a similar process, where multiple classifiers are trained to iteratively learn and generate additional training labels. Zhang et al. [10] proposed an enhanced multi-modal co-training algorithm for semi-supervised emotion recognition.

With similar motivation, our work implements self-training to leverage information from unlabeled data. However, instead of using the predicted class of unlabeled samples as ground-truth, we use the predicted class distribution as the true distribution in our model. This allows the model to partially adapt to the expression data of different intensities.

C. Methods with Noisy and Inconsistent Labels

Presence of noisy and inconsistent data can easily overfit a CNN model. Expression annotations are very subjective and datasets created from crowd-sourcing are often inconsistent and noisy. To avoid the risk of learning from the noisy data, a multi-task network is proposed in [25] to jointly learn the cleaning of the noisy annotation and classifying to the correct class. Li et al. [26] proposed a unified distillation framework to learn from noisy labels by leveraging a knowledge graph.

Another way to address inconsistently labeled data is dataset resampling, i.e., minimizing the weighted loss during training using sample weights. Jiang et al. [15] proposed MentorNet to learn the weights of the examples using long short term memory based on the training loss. The effect of each training points on the CNN model's performance was exploited in [27] by analyzing the change of model parameters due to removal of a data point. Data dropout [13] was proposed to optimize the training data by removing the unfavorable or bad training samples. The idea behind instance-based transfer learning [14] was to use a pre-trained

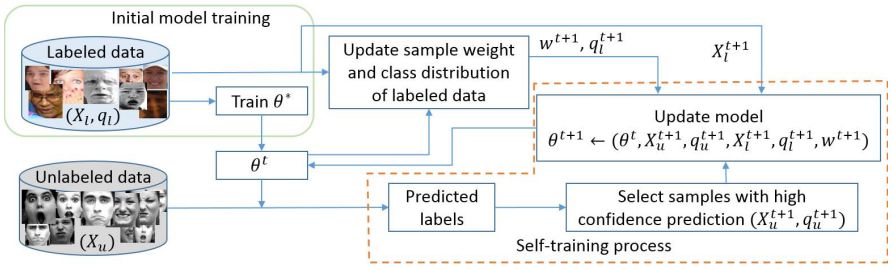


Fig. 1: Workflow of the proposed expression recognition method.

model to estimate the influence of target domain samples and to optimize the training data by removing samples that will lower the model performance.

Our approach assigns weight to each labeled sample which reflects its importance during model update. In contrast to other methods in the literature, the sample weight in our approach is modified in each iteration based on the training and validation losses. Our model allows both down-weighting the hard examples with high loss (usually noisy annotation) and up-weighting the easy examples with clean annotation. Thus, our model is robust to outliers and noisy data.

III. Proposed Method

The block diagram of the proposed work is shown in Figure 1. The idea behind our work is to train an initial model for expression classification, which is afterward used to classify the unlabeled data. A few samples from unlabeled data with high prediction scores are selected (with corresponding predictions as ground truth) and treated as train data for further model update. The predicted class probabilities for the train data are considered to update the ground truth vector of the train data. In addition, each training sample is assigned with a weight according to its importance in updating the model parameters. Thus, our framework has four main parts: (1) *initial model training*: obtaining the model parameters for fairly accurate expression classification, (2) *exploiting unlabeled data for model update*: leveraging the unannotated data to improve the model performance by self-training, (3) *ground-truth distribution correction for train data (label update)*: allowing the model to adapt to different emotion intensities while maintaining certain confidence in prediction, and (4) *assigning sample importance*: computing the sample weights and ignoring the inappropriate samples during model update. We proceed to discuss the details of these steps in this section.

A. Initial Model Training

The task of an expression recognition framework is to correctly predict the emotion class or the emotion class probabilities. Given the sample and label pair $(\mathbf{x}_i, y_i)$, the classification network learns to accurately predict class probabilities $p(k|\mathbf{x}_i, \theta)$, where θ are the network parameters and $k \in \{1, 2, ..., K\}$ represents K classes. For a soft-max layer, we have $p(k|\mathbf{x}_i, \theta) = \frac{\exp(z_k)}{\sum_{j=1}^{K} \exp(z_j)}$, where z_j are the unnormalized log probabilities. In supervised learning, the ground-truth distribution $q(k|\mathbf{x}_i)$ is used to train the network parameters (θ) by minimizing the cross-entropy loss function

$$\mathcal{L} = \sum_{\mathbf{x}_i \in X_l} f_i(\theta) = - \sum_{\mathbf{x}_i \in X_l} \sum_{k=1}^{K} \log(p(k|\mathbf{x}_i, \theta)) q(k|\mathbf{x}_i). \quad (1)$$

Here $f_i(\theta)$ denotes the loss incurred by $\mathbf{x}_i$ and X_l represents the set of training samples. Usually, one-hot encoding is used for classification models, which takes the form $q(y_i|\mathbf{x}_i) = 1$ and $q(k|\mathbf{x}_i) = 0$ for all $k \neq y_i$. For clarity, we drop the subscript and use $\mathbf{x}$ instead of $\mathbf{x}_i$ to denote a sample.

The initial model (θ^*) is utilized to predict pseudo class probabilities pertaining to unlabeled data during the self-training process. Thus, the initial model needs to be fairly accurate to avoid erroneous predictions, which in turn will result in a poorer model due to error accumulation. We trained the initial model using the labeled data for sufficient epochs depending upon the dataset until adequate performance is achieved.

Label Smoothing: Unlike object classification, expression categories are highly related, interconnected, as well as can occur simultaneously. For instance, multiple expression classes coexist during emotion transition or mixed emotions. As the expression datasets contain expressive images of various intensities, it would be best to train a classifier with normalized emotion intensity vectors as the ground truth. However, such datasets do not include the emotion intensity information. Training a model with one-hot vector usually results in overfitting the model. Therefore, the presence or absence of multiple objects can be formulated with one-hot vectors, whereas the expression recognition model should consider soft class assignment vectors.

Strong supervision with one-hot encoding over-fits the model on the training data and fails to generalize for the unseen data. Label smoothing [12] provides a solution to this situation by replacing the one-hot encoding with a smooth distribution. Though it renders low confidence prediction, the model becomes more regularized and adaptable to unseen data. We implemented the label smoothing as,

$$q'(k|\mathbf{x}) = \begin{cases} 1 - \epsilon, & k = y \\ \frac{\epsilon}{K-1}, & k \neq y, \end{cases} \quad (2)$$

where $\epsilon \in [0, 1]$ is the label smoothing hyperparameter. While setting $\epsilon = 0$ refers to one-hot encoding, setting ϵ a large value might result in learning a poor performing model.

B. Exploiting Unlabeled Data for Model Update

The proposed semi-supervised method uses a fraction of the unlabeled data along with the labeled samples to update the network. We used a self-training procedure inspired by [16], where the class labels of the unlabeled data are estimated using the network predictions. Unlike in [16], we use the predicted probability distribution as the ground-truth distribution.

Suppose X_u denotes the unlabeled data, q_l stands for ground-truth distribution of labeled data, and p_u denotes the network prediction probabilities for unlabeled data. Note that the initial model (θ^*) is trained with X_l and q_l. Afterward, we update the model parameters iteratively using a portion of data from both X_l and X_u simultaneously. The unlabeled samples are selected from the pool of unlabeled data predicted with high confidence scores ($\widehat{X_u}$), as suggested in [11]. Instead of using all the high confidence samples, we select a few samples randomly from $\widehat{X_u}$ in each iteration to avoid over-fitting the model.

Maintaining a proper balance between the number of labeled and unlabeled data is crucial to avoid error accumulation in self-training process. In our implementation, we randomly replace a fraction of X_l (typically $1-10\%$ of number of labeled samples in X_l) with the unlabeled data from $\widehat{X_u}$ in each epoch. After the t-th update, we obtain the prediction scores using θ^t for both X_l and X_u, denoted by p_l^t and p_u^t, respectively. We obtain the unlabeled data with high prediction scores using a threshold value (τ), given by

$$\widehat{X_u^t} = \{\mathbf{x} | \mathbf{x} \in X_u \text{ and } \max_k p_u^t(k|\mathbf{x}) > \tau\}; \quad \widehat{X_u^t} \subset X_u. \tag{3}$$

From the pool of samples in $\widehat{X_u^t}$, we select a few randomly and replace with the random samples from X_l. Thus, the train data for $(t+1)$-th iteration becomes $X^{t+1} = \{X_l^{t+1}, X_u^{t+1}\}$, where $X_l^{t+1} \subset X_l$ and $X_u^{t+1} \subset \widehat{X_u^t}$ are selected randomly. In our implementation, X_u^{t+1} consists of equal amount of data from each class to avoid the trivial solution of predicting the dominant class for unlabeled data in successive iterations. We empirically found that τ in the range $(0.7, 0.95)$ is suitable for different datasets, as it promises dominant class structure while adopting to moderate expression intensities.

C. Label update

The ground-truth distribution of the labeled data is cautiously modified to make the model adapt to expression samples of different intensities. It is carried out by updating the class distribution based on the model prediction. We apply label smoothing on the model predictions for the labeled data and utilize them for further parameter update, which we call *successive label smoothing*. In contrast, the predicted class probabilities for the unlabeled data are directly used for model training.

1) Updating q_l^t: By performing successive label smoothing on labeled data, the model adapts to the expression samples of various intensities. However, incorrect predictions on X_l can accumulate error and reduce model performance in subsequent iterations. It is important to *maintain the prediction confidence of labeled data*, while learning the necessary information from the unlabeled data. In order to achieve that, we scrutinize p_l^t after each iteration and force the model to rectify its prediction errors on X_l. We maintain the confidence of the model by forcing the model to correctly predict the expression class with a prediction score greater than certain threshold (α), which is given by,

$$q_l^{t+1}(k|\mathbf{x}) = \begin{cases} p_l^t(k|\mathbf{x}), & \text{if } \{ \max_k p_l^t(k|\mathbf{x}) > \alpha \\ & \text{and } \operatorname{argmax}_k p_l^t(k|\mathbf{x}) = y \ \} \\ g(p_l^t(k|\mathbf{x})), & \text{otherwise} \end{cases} \tag{4}$$

$$\text{where} \quad g(p_l^t(k|\mathbf{x})) = \begin{cases} \alpha, & \text{if } k = y \\ \frac{1-\alpha}{K-1}, & \text{if } k \neq y. \end{cases} \tag{5}$$

Essentially, this means that we use the prediction probabilities as true distribution if the model predicts the correct expression class (y) with a probability above threshold α. The failure cases are assigned with ground-truth distribution with label smoothing (as shown in equation (5)). Thus, α determines the minimum probability that can be assigned to the correct class.

2) Updating q_u^t: The ground-truth distribution of unlabeled data is updated to its predicted probabilities, i.e., $q_u^{t+1}(k|\mathbf{x}) = p_u^t(k|\mathbf{x})$. Thus, the model sees the unlabeled data with corresponding updated ground-truth distribution after each update.

The model is updated from θ^t to θ^{t+1} in a supervised manner using $\{X_l^{t+1}, X_u^{t+1}\}$ and the corresponding updated ground-truth probabilities $\{q_l^{t+1}, q_u^{t+1}\}$. This forces the model to generate closely similar probabilities every time it accepts a particular sample as input. Intuitively, the model will train itself to correctly classify the supervised data, while incorporating the variations from the unlabeled data into the model. Here the parameter α controls the prediction confidence of labeled data. Choosing the value of $\alpha \approx 1$ restricts the model to learn for definite dominant expressions. However, as discussed before, data of different expression-intensities might have a similar label and the value of α should be in the range 0.6 to 0.9, in order to allow the model to adapt to different intensities.

D. Assigning Sample Importance

The presence of wrongly annotated or inappropriate training data hinders the model from obtaining optimal parameters and negatively affects the model performance. Such samples are assigned with low *sample importance* or *sample weight*, thereby avoiding their effect on the model update. This is carried out by penalizing the loss incurred by the labeled training samples (X_l) by the corresponding sample weight ($0 \leq w_i \leq 1$).

The loss function used at tth iteration is given by

$$\mathcal{L} = \sum_{\mathbf{x}_i \in X_l^t} w_i^t f_i(\theta^t) + \sum_{\mathbf{x}_i \in X_u^t} f_i(\theta^t). \tag{6}$$

Comparing with equation (1), here we minimize the weighted loss, where the sample weights are learned using the training loss. As we assume the selected unlabeled data is clean, the corresponding loss is not penalized. Further, we assume that all the labeled samples are equally important

initially, i.e., $w_i^0 = 1, \forall \mathbf{x}_i \in X_l$. After training the initial model θ^*, the sample weights are updated in every iteration t based on the loss values incurred by individual sample. Further, we use the overall loss obtained for the validation set, $\mathcal{L}_{val}^t = \sum_{\mathbf{x}_i \in X_{val}} f_i(\theta^t)$, to decide the worthiness of the sample weights. Notice that the validation loss is computed without the sample weights.

We update w_i^t if the trend of average validation loss starts increasing, i.e., $\overline{\mathcal{L}_{val}^t} > \overline{\mathcal{L}_{val}^{t-1}}$, where $\overline{\mathcal{L}_{val}^t} = \frac{1}{T}\sum_{t=t-T+1}^{t} \mathcal{L}_{val}^t$ is the average validation loss over previous T iterations. We used $T = 3$ in all experiments. The weight update is carried out by

$$w_i^{t+1} \leftarrow w_i^t - \Delta w_i^t; \text{ where} \tag{7}$$

$$\Delta w_i^t = c.\ \tanh\Big(\frac{f_i(\theta^t) - f_{\text{avg}}(\theta^t)}{\sigma}\Big). \tag{8}$$

Here $f_{\text{avg}}(\theta^t)$ is the average training loss at tth iteration, whereas $0 \leq c \leq 1$ and $\sigma > 0$ are scalars that control the weight update in consecutive iterations. Note that Δw_i^t can take both positive and negative values, subsequently decreasing or increasing the sample weights. Δw_i^t takes positive values if the loss incurred due to a sample is higher than the average loss, and vice versa. We further clip the sample weights in the range $[0, 1]$ to avoid unpleasant computations.

IV. Experiments and Results

A. Implementation Details

Datasets

Both *in-the-lab* (CK+ [28], RaFD [29]) and *in-the-wild* (RAF [1], AffectNet [30]) datasets are used in our experiments. The *in-the-wild* datasets contain data collected from uncontrolled environments and thus cover real-world expressions with various facial poses, illuminations, emotion intensity, occlusion, and other factors. Whereas the *in-the-lab* datasets are mostly posed in a controlled environment and contain exaggerated expressions of frontal faces.

All these datasets (except CK+) contain annotation for seven expressions, namely: anger, disgust, fear, happy, neutral, sad, and surprise. CK+ [28] contains 327 sequences annotated with 7 expression labels (six basic emotions and contempt). The contempt class is excluded in our experiments as we aim for cross-database evaluation and comparison purposes [17]. Following the literature [17], [6], [7], we use the 10-fold cross-validation while using the first frame as neutral and the last three frames of each sequence of CK+ as the corresponding expression label. The frontal faces (1407 samples) are used from RaFD dataset for our experiments. We conduct five-fold cross-validation for RaFD evaluation.

AffectNet and RAF are collected from the internet by using certain emotional terms in various search engines. RAF [1] contains manually annotated images, out of which 12271 are listed in *RAF-train* and 3068 samples in *RAF-test*. AffectNet [30] has around 400000 labeled data in *AffectNet-train* and 5000 samples in *AffectNet-test*. The agreement between two annotators in AffectNet is found to be 60%, which explains the complexity and subtlety of the expressions in this dataset. Following the experimental settings of [31] and [32], we only use the seven classes for these datasets.

TABLE I: Comparison of average classification accuracy for different combinations of self-training (ST) and sample weight assignment (SW). Here the unlabeled samples come from the test-split of the same dataset. The best accuracy is reported in **bold**.

Datasets	without ST and SW	with ST only	with SW only	with both ST and SW
CK+	99.19	99.19	99.27	**99.43**
RaFD	99.29	99.5	99.49	**99.71**
RAF	83.44	83.6	83.83	**84.61**
AffectNet	55.48	56.62	57.31	**57.37**

TABLE II: Performance improvement on RAF and AffectNet datasets by using *in-the-lab* datasets as unlabeled data source. (using both ST and SW)

(a) RAF dataset

Unlabeled data source	Accuracy
RAF-test	84.61
CK+	84.81
RaFD	**84.94**

(b) AffectNet dataset

Unlabeled data source	Accuracy
AffectNet-test	57.37
CK+	61.54
RaFD	**62.25**

Preprocessing

Preprocessing steps involve face detection (using MTCNN [33]) and face alignment, in order to position both eyes at a fixed distance parallel to the horizontal axis. The training set is augmented using slight zooming, horizontal flipping, less than 10% vertical and horizontal shifting, as well as rotating the images randomly in the range of ± 10 degrees. For both these in-the-wild datasets, we use the aligned images provided by the dataset developers. We normalize the training data to zero mean and unit variance in all experiments.

Network

We use the pre-trained VGG-Face model proposed by Parkhi *et al.* [21] initially introduced for face recognition. It consists of thirteen convolutional layers followed by two fully connected layers. The aligned faces are re-sized to 224×224 resolution and feed to the CNN models. We set the softmax layer to the number of expression classes (in our case seven: anger, disgust, fear, happy, neutral, sadness, and surprise). The initial model weights are obtained from the pretrained VGG-Face model, achieving knowledge transfer. We use Adam optimizer with the suggested weights $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a learning rate of 1e–4 in our model. All models are trained using a batch size of 32. The experiments are carried out using NVIDIA 1080 GPU with CUDA to improve the speed.

Parameter Settings

We conduct several experiments by varying ϵ from $0.05 - 0.2$, and varying both τ and α in the range $0.6 - 0.9$ to select the suitable values of corresponding parameters. Empirically we find $\epsilon = 0.05$, $\tau = 0.9$ and $\alpha = 0.7$ to be adequate for all the experiments, irrespective of the database type and mode of evaluation. The weight update parameters are also set empirically to $c = 0.3$ and $\sigma = 2$. In all our experiments, we replace 2% of training data with high confidence unlabeled samples. We use 10% of the train data as the validation data in all experiments.

TABLE III: Cross-dataset accuracy for seven expression classes. Note that we utilized the test dataset as the unlabeled data.

(a) without ST and SW

Train \ Test	CK+	RaFD	RAF	AffectNet
CK+	-	75.83	46.15	38.56
RaFD	80.9	-	48.34	39.75
RAF	72.49	70.5	-	42.2
AffectNet	82.92	84.79	63.03	-

(b) with ST only

Train \ Test	CK+	RaFD	RAF	AffectNet
CK+	-	76.04	45.79	38.82
RaFD	82.76	-	48.96	40.58
RAF	71.35	84.86	-	43.17
AffectNet	86.48	92.18	64.73	-

(c) with ST and SW

Train \ Test	CK+	RaFD	RAF	AffectNet
CK+	-	86.56	48.45	39.87
RaFD	87.54	-	49.54	41.63
RAF	73.26	91.61	-	44.08
AffectNet	87.37	94.02	64.65	-

B. Performance on Different Datasets

Using samples from test set as unlabeled data

The average accuracy obtained for different datasets by the proposed method for self-training (ST) and/or sample weight assignment (SW) is presented in Table I. We report the accuracy for four different conditions based on the utilization of ST and SW.

As can be seen from Table I, the accuracy of the framework without ST and SW is the lowest for all datasets. Importantly, one can notice performance improvements by using either of ST or SW alone. This clearly demonstrates the effectiveness of ST and SW in emotion recognition. When ST and SW are applied together, we obtained the best performance in all datasets. This proves the effectiveness of combining the two approaches. As can be observed from Table I, we obtain close to perfect performances in both the *in-the-lab* datasets. However, the accuracy for the *in-the-wild* datasets is comparatively very poor. We obtain an accuracy of 57.37% only in AffectNet dataset.

Using *in-the-lab* datasets as unlabeled data

We conduct experiments to improve the performance of *in-the-wild* datasets by leveraging the information from *in-the-lab* datasets. Therefore, we report the performance of RAF and AffectNet in two conditions: (1) the test split of the same dataset is used as the source for unlabeled data, and (2) one of the *in-the-lab* dataset is utilized as the unlabeled data source. For example, to evaluate the performance of AffectNet, we use *AffectNet-train* for training the model while evaluating with *AffectNet-test*. However, the unlabeled data used during model training can come from either (1) *AffectNet-test* or (2) CK+ or RaFD. Both cases are reported in Table II. Here all the results are obtained using both ST and SW.

From Table II, we observe performance improvement in both RAF and AffectNet by using the *in-the-lab* datasets as the source of unlabeled data. We obtain 84.94% and 62.25% accuracy on RAF and AffectNet respectively. The model performance is especially improved by 5% for AffectNet by using unlabeled samples from RaFD. The presence peak intensity samples in *in-the-lab* datasets might be the reason behind this performance boost. Our model selects the high confidence unlabeled samples with the corresponding predictions as true distribution. This assumption is mostly true for *in-the-lab* datasets as the chances of peak expressive images getting classified into correct classes is high with a fair performing model. Thus, the unlabeled samples selected in the process drives the model to accurately present the expression prototypes while learning the variations of the in-the-wild labeled data.

The classification results on some AffectNet samples are presented in Figure 2. One can observe that the predictions for different correctly classified samples (see Figure 2a) resemble the emotion intensities. The increase in prediction scores for *happy* and *sad* classes are demonstrated in the top row of Figure 2a. We believe that the ability of the model to adapt to the sample intensity is due to two factors: (1) successive label smoothing, and (2) using the predicted class probabilities as true distribution for unlabeled samples.

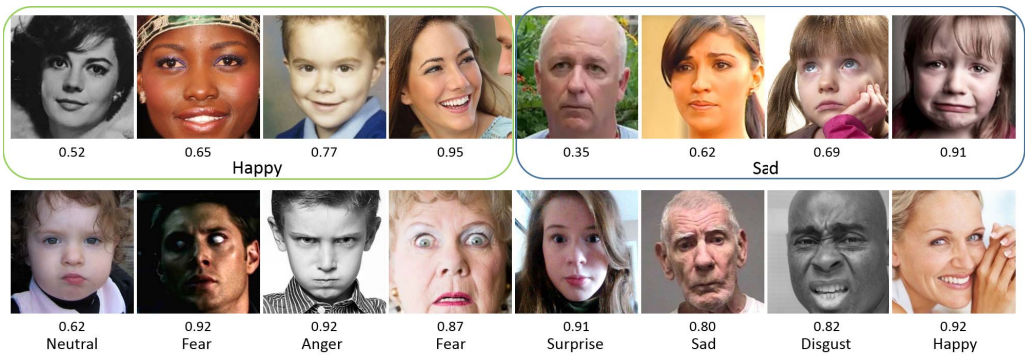
Figure 2b and 2c displays the samples wrongly classified by our model. However, the predicted classes in Figure 2b are more appropriate than the ground-truth annotation. Similarly, Figure 2c represents the images with blended emotions, where the model predicts the second dominant class. The presence of noisy annotation in test split (*AffectNet-test* and *RAF-test*) makes the evaluation of such datasets more difficult.

The effectiveness of the SW is demonstrated in Figure 3. Our model automatically assigns lower sample weights to the noisy samples as shown in the top row in Figure 3. The samples wrongly annotated with *happy* class are also assigned with low sample weight. This demonstrates the robust training of the model against inconsistently labeled data. In addition to the noisy data, the model assigns low weight to the difficult samples with lower emotion intensity.

C. Cross-Database Evaluation

Table III reports the performance of seven class classification for cross-dataset evaluation. We present the results in the ascending order of dataset size (CK+ <RaFD <RAF <AffectNet). One interesting trend that can be observed from Table III is that the models trained with large datasets perform well on small ones. This has also been observed in [17] and [18], where this trend is considered as the effect of fewer variations in training data due to low sample size. Furthermore, the occlusion, pose and illumination variations are absent in-the-lab datasets; thus, it is difficult to train the complex model architecture without causing significant overfitting.

The effect of ST and SW is also validated by the cross-database performance as shown in Table III. As can be observed, the performance of the model improves by using ST, which further improves by using both ST and SW. For instance, the AffectNet $\rightarrow$ RaFD model performance is improved from 84% to 94% when both ST and SW are implemented. In the same way, RaFD $\rightarrow$ CK+, RAF $\rightarrow$ RaFD, and RAF $\rightarrow$ AffectNet sees an performance gain of 7%, 20%, and 2% respectively among others.



(a) Correctly recognized samples from AffectNet dataset. Prediction scores are provided below the images which closely resemble the emotion intensities.

(b) Samples for which model predication is more appropriate than the ground-truth.

(c) Mis-classified samples. The predicted class is the second dominant emotion for the blended expression samples.

Fig. 2: Classification examples from AffectNet dataset.

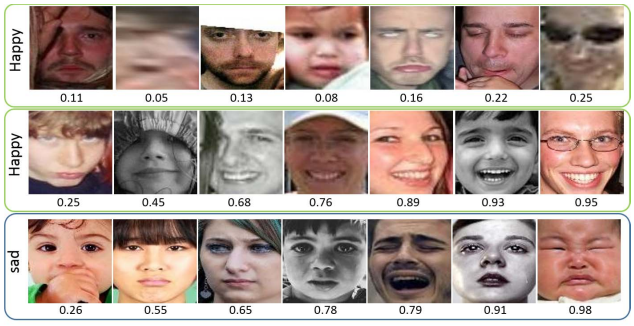


Fig. 3: Training samples from RAF data. The value below the images depicts the sample importance learned by our framework. (*Top row*: Noisy annotations for *happy* class. Low sample importance neglects the effect of noisy inconsistent samples on model learning. *Middle and Bottom row*: The assigned sample weights for *happy* and *sad* class in ascending order.)

D. Comparison with Other Methods

We compare our results with recent CNN-based methods and state-of-the-art results. All the results reported here are average classification accuracy for the seven expression classes. Our results outperform the state-of-the-art in both the *in-the-lab* datasets. As these datasets contain frontal faces with peak expressions, the performance of the framework is close to perfect. However, the accuracy drops drastically when *in-the-wild* datasets are considered. The proposed method outperforms the state-of-the-art (in Table VIb) at a margin of 4% for AffectNet dataset. However, the performance achieved in IPA2LT [19] is better than the proposed method for RAF dataset (see Table VIa). Note that samples from both AffectNet and RAF are combined during the training process in [19], which might provide the model an upper hand.

TABLE IV: Comparison of classification accuracy on CK+.

Methods	Validation settings	Accuracy
LOMo [34]	7 class	95.1
IACNN [7]	7 class	95.37
BDBN[17]	6 class [1]	96.7
DTAGN [4]	7 class	97.25
PPDN [5]	6 class	97.3
Facenet2expnet [6]	6 class	98.6
PPDN [5]	7 class	99.3
Proposed	7 class	**99.43**

[1]Eight-fold cross-validation is performed.
[2]Trained with both AffectNet and RAF train set

TABLE V: Comparison of classification accuracy on RaFD.

Methods	Accuracy
BAE-BNN-3[35]	96.93
TLCNN+FOS[36]	97.75
Carcagni *et al.*[37]	98.5
Proposed	**99.71**

TABLE VI: Performance comparison on RAF and AffectNet.

(a) RAF dataset

Methods	Accuracy
CAKE [32]	68.9
DLP-CNN [1]	74.2
Vielzeuf *et al.* [38]	80
PG-CNN [31]	83.27
IPA2LT [19][2]	**86.77**
Proposed	84.94

(b) AffectNet dataset

Methods	Accuracy
PG-CNN [31]	55.33
IPA2LT [19][2]	57.31
CAKE [32]	58.1
AlexNet [30]	58
Proposed	**62.25**

V. Conclusions

In this paper, we proposed a unified semi-supervised learning model for expression recognition which uses (a) self-training, exploiting information from abundant unlabeled data, (b) successive label smoothing, allowing the model to adapt to the emotion intensities and perform well on low intensity data, as well as (c) sample weight assignment, in order to avoid learning from noisy and inconsistent data. We experimentally validated the performance improvement with self-training and sample weight assignment. Experiments conducted on four public datasets indicate (i) excellent model performance out-performing state-of-the-art for most datasets, (ii) ability to learn different expression intensities, even when trained with categorical samples, (iii) ability to learn from inconsistent and noisy data, (iv) significant performance improvement on *in-the-wild* datasets by leveraging the information from the *in-the-lab* datasets, and (v) large performance gains in cross-database performance.

References

[1] S. Li, W. Deng, and J. Du, "Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild," in *CVPR*. IEEE, 2017, pp. 2584–2593.

[2] Y. Zhou, H. Xue, and X. Geng, "Emotion distribution recognition from facial expressions," in *ACMMM*. ACM, 2015.

[3] A. Dantcheva, P. Bilinski, H. T. Nguyen, J.-C. Broutart, and F. Bremond, "Expression recognition for severely demented patients in music reminiscence-therapy," in *EUSIPCO*, 2017.

[4] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim, "Joint fine-tuning in deep neural networks for facial expression recognition," in *ICCV*. IEEE, 2015, pp. 2983–2991.

[5] X. Zhao, X. Liang, L. Liu, T. Li, Y. Han, N. Vasconcelos, and S. Yan, "Peak-piloted deep network for facial expression recognition," in *ECCV*. Springer, 2016, pp. 425–442.

[6] H. Ding, S. K. Zhou, and R. Chellappa, "Facenet2expnet: Regularizing a deep face recognition net for expression recognition," in *Int. Conf. on Automatic Face & Gesture Recognition (FG)*. IEEE, 2017, pp. 118–126.

[7] Z. Meng, P. Liu, J. Cai, S. Han, and Y. Tong, "Identity-aware convolutional neural network for facial expression recognition," in *Int. Conf. on Automatic Face & Gesture Recognition (FG)*. IEEE, 2017, pp. 558–565.

[8] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, "Deep learning for emotion recognition on small datasets using transfer learning," in *ACM Int. Conf. on multimodal interaction*. ACM, 2015, pp. 443–449.

[9] S. Happy, A. Dantcheva, and F. Bremond, "A weakly supervised learning technique for classifying facial expressions," *Pattern Recognition Letters*, vol. 128, pp. 162–168, 2019.

[10] Z. Zhang, F. Ringeval, B. Dong, E. Coutinho, E. Marchi, and B. Schüller, "Enhanced semi-supervised learning for multimodal emotion recognition," in *ICASSP*. IEEE, 2016, pp. 5185–5189.

[11] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," in *WACV Workshops*, 2005.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016, pp. 2818–2826.

[13] T. Wang, J. Huan, and B. Li, "Data dropout: Optimizing training data for convolutional neural networks," in *Int. Conf. on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2018, pp. 39–46.

[14] T. Wang, J. Huan, and M. Zhu, "Instance-based deep transfer learning," in *WACV*. IEEE, 2019, pp. 367–375.

[15] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *ICML*, 2018, pp. 2309–2318.

[16] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on Challenges in Representation Learning, ICML*, vol. 3, 2013, p. 2.

[17] P. Liu, S. Han, Z. Meng, and Y. Tong, "Facial expression recognition via a boosted deep belief network," in *CVPR*, 2014, pp. 1805–1812.

[18] A. T. Lopes, E. de Aguiar, A. F. De Souza, and T. Oliveira-Santos, "Facial expression recognition with convolutional neural networks: coping with few data and the training sample order," *Pattern Recog. (PR)*, vol. 61, pp. 610–628, 2017.

[19] J. Zeng, S. Shan, and X. Chen, "Facial expression recognition with inconsistently annotated datasets," in *ECCV*, 2018, pp. 222–37.

[20] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *CVPR*. IEEE, 2014, pp. 1717–1724.

[21] O. M. Parkhi, A. Vedaldi, A. Zisserman *et al.*, "Deep face recognition." in *BMVC*, vol. 1, no. 3, 2015, p. 6.

[22] A. Coates and A. Y. Ng, "Learning feature representations with k-means," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 561–580.

[23] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *NIPS*, 2015, pp. 3546–3554.

[24] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *ICLR*, 2016.

[25] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie, "Learning from noisy large-scale datasets with minimal supervision," in *CVPR*, 2017, pp. 839–847.

[26] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li, "Learning from noisy labels with distillation," in *ICCV*, 2017, pp. 1910–1918.

[27] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *ICML*, 2017, pp. 1885–1894.

[28] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in *CVPR Workshops*. IEEE, 2010, pp. 94–101.

[29] O. Langner, R. Dotsch, G. Bijlstra, D. H. Wigboldus, S. T. Hawk, and A. Van Knippenberg, "Presentation and validation of the radboud faces database," *Cognition and emotion*, vol. 24, no. 8, pp. 1377–1388, 2010.

[30] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "Affectnet: A database for facial expression, valence, and arousal computing in the wild," *IEEE Trans. on Affective Comput.*, 2017.

[31] Y. Li, J. Zeng, S. Shan, and X. Chen, "Patch-gated cnn for occlusion-aware facial expression recognition," *ICPR*, 2018.

[32] C. Kervadec, V. Vielzeuf, S. Pateux, A. Lechervy, and F. Jurie, "Cake: Compact and accurate k-dimensional representation of emotion," in *BMVC Workshop*, 2018.

[33] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Proces. Let.*, vol. 23, no. 10, pp. 1499–1503, 2016.

[34] K. Sikka, G. Sharma, and M. Bartlett, "Lomo: Latent ordinal model for facial analysis in videos," in *CVPR*, 2016, pp. 5580–5589.

[35] W. Sun, H. Zhao, and Z. Jin, "An efficient unconstrained facial expression recognition algorithm based on stack binarized auto-encoders and binarized neural networks," *Neurocomputing*, vol. 267, pp. 385–395, 2017.

[36] Y. Zhou and B. E. Shi, "Action unit selective feature maps in deep networks for facial expression recognition," in *IJCNN*. IEEE, 2017, pp. 2031–2038.

[37] P. Carcagnì, M. Coco, M. Leo, and C. Distante, "Facial expression recognition and histograms of oriented gradients: a comprehensive study," *SpringerPlus*, vol. 4, no. 1, p. 645, 2015.

[38] V. Vielzeuf, C. Kervadec, S. Pateux, A. Lechervy, and F. Jurie, "An occam's razor view on learning audiovisual emotion recognition with small training sets," in *ICMI*. ACM, 2018, pp. 589–593.