Co-Incrementation: Combining Co-Training and Incremental Learning for Subject-Specific Facial Expression Recognition

Jordan Gonzalez[1], Thibault Geoffroy[1], Aurelia Deshayes[2] and Lionel Prevost[1]
[1]*Learning, Data and Robotics (LDR) Lab, ESIEA, Paris, France*
[2]*Laboratoire d'Analyse et Mathématiques Appliquées (LAMA), UPEC, Créteil, France*

Keywords: Incremental Learning, Semi-Supervised Learning, Co-Training, Random Forest, Emotion Recognition.

Abstract: In this work, we propose to adapt a generic emotion recognizer to a set of individuals in order to improve its accuracy. As this adaptation is weakly supervised, we propose a hybrid framework, the so-called co-incremental learning that combines semi-supervised co-training and incremental learning. The classifier we use is a specific random forest whose internal nodes are nearest class mean classifiers. It has the ability to learn incrementally data covariate shift. We use it in a co-training process by combining multiple view of the data to handle unlabeled data and iteratively learn the model. We performed several personalization and provided a comparative study between these models and their influence on the co-incrementation process. Finally, an in-depth study of the behavior of the models before, during and after the co-incrementation process was carried out. The results, presented on a benchmark dataset, show this hybrid process increases the robustness of the model, with only a few labeled data.

1 INTRODUCTION

These last decades, the field of automated emotion recognition has dramatically grown. New solutions, mainly based on machine learning algorithms, have been developed. New data sets have been shared within the research community and many emerging applications are starting to mature in various fields like video gaming, education, health, medical diagnosis, etc. Nevertheless, many challenges remain, like face lightning or face pose (Sariyanidi et al., 2014).

In this paper, we address the challenge of identity bias. The individual variability between people has a direct consequence on the recognition of emotions (Senechal et al., 2010). On the morphological level, the shape and "texture" (skin, wrinkles) of the face differ according to different factors (gender, age, etc.). On the behavioral level, each subject has his own way of expressing emotion, depending on his introverted or extroverted personality.

It is well known that building accurate classifiers involves gathering labeled data. Due to time and cost constraints, it seems irrational to expect labeling hundreds or even thousands of data that would be necessary to train today's models in a supervised way. Moreover, labeling (potentially many) new affective data corresponding to new subjects seems too costly.

To solve this issue, one solution is to personalize generic classifiers. In detail, these classifiers are trained on large-scale datasets containing many subjects. Obviously, regardless of the dataset size, it is highly unlikely to capture all of the inter-subject variability in terms of morphology and behavior. Consequently, these generic "omni-subjects" classifiers will perform the best they can, given the biases described above. The idea behind personalization is to adapt these classifiers to a given set of subjects (multi-subjects approach), or even to one particular subject (mono-subject approach). Researchers in areas like handwriting recognition (Oudot et al., 2004) and speech recognition (Meng et al., 2019) got to work on this "variability problem". To overcome this challenge, they built self-adaptation (incremental) algorithms. The general idea of these algorithms is to continuously adapt a generic ("world") model to personalize it on one single user, by using language constraints, for example. Unfortunately, for emotion recognition in images, such rules do not exist. That is why we need to turn to other solutions.

In the field of semi-supervised learning, the co-training algorithm (detailed in the next section) uses several models, trained on labeled data, to predict the class (also called pseudo-label) of unlabeled data. Then, it add these data to labeled set and retrain the models on this augmented dataset. This process has a high cost in terms of computational time.

Gonzalez, J., Geoffroy, T., Deshayes, A. and Prevost, L.
Co-Incrementation: Combining Co-Training and Incremental Learning for Subject-Specific Facial Expression Recognition.
DOI: 10.5220/0011635200003411
In *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2023)*, pages 270-280
ISBN: 978-989-758-626-2; ISSN: 2184-4313

The main proposal of this work is to combine co-training with incremental learning in order to avoid this issue. Initial models are generic ones, trained on a large set of subjects. They are used to predict the pseudo-label of new data, corresponding to new subjects. After pseudo-labeling these new samples, we apply incremental learning technics to adapt the models. Therefore, these models are no more generic, but personalized to new subjects. We called this process co-incrementation learning. One of its main advantage is to reduce drastically the computing time while improving the recognition accuracy on new subjects, thus reducing the identity bias.

The rest of the article is as follows. Next section will present the incremental learning field with a particular focus on random forest (RF)-based algorithms and detail the co-training process. Section 3 is devoted to the data and the feature extraction process. Section 4 presents in detail the nearest-class mean forest (NCMF), how it differs from classical RF and the way it can learn incrementally. In section 5, we detail the original co-incrementation algorithm that combines incremental NCMF with co-training. Then, we present in section 6 results obtained on generic models (before adaptation) and after co-incrementation on specific chunks. Finally, we conclude in section 7.

2 RELATED WORKS

Automatic Facial Emotion Recognition (FER) has received wide interest in a variety of contexts, especially for the recognition of action units, basic (or compound) emotions and affective states. Although considerable effort has been made, several questions remain about which cues are important for interpreting facial expressions and how to encode them. Affect recognition systems most often aim to recognize the appearance of facial actions, or the emotions conveyed by those actions (Sariyanidi et al., 2014). The former are generally based on the Facial Action Coding System (FACS)(Ekman, 1997). The production of a facial action unit has a temporal evolution, which is typically modeled by four temporal segments: neutral, onset, apex, and offset (Ekman, 1997). Among them, the neutral is the phase with no expression and no sign of muscle activity; the apex is a plateau where the maximum intensity usually reaches a stable level.

As seen before, identity bias results in performance losses on generic learning models. Strategies for grouping individuals by common traits such as gender, weight, or age and personalizing models on these groups have already shown promising results in a wide range of areas such as activity recognition (Chu et al., 2013) (Kollia, 2016) (Yang and Bhanu, 2011). However, quite often the strategy used consists in personalizing one model per user since it ensures better results. This can quickly become complex when the number of subjects increases or when the number of collected data per subjects keeps small. In the field of emotion recognition, different solutions to this challenge have been considered, personalization methods being the most promising (Chu et al., 2013) (Yang and Bhanu, 2011).

One of the main characteristics of incremental techniques is the ability to update models using only recent data. This is often the only practical solution when it comes to learning data "on the fly" as it would be impossible to keep in memory and re-learn from scratch every time new information becomes available. This type of technique holds promise for personalizing models to individuals. It has been demonstrated that Random forests (RF) (Breiman, 2001), in addition to their multi-class nature and ability to generalize, have also the ability to increment in data and classes (Denil et al., 2013) (Hu et al., 2018) (Lakshminarayanan et al., 2014) . Besides, Random forest models have been used successfully for personalization (Chu et al., 2013) (Kollia, 2016) (Yang and Bhanu, 2011). Nearest class mean forests derived from RF, have demonstrated to be able to outperform RF performance and allow an easy way to perform incrementation (Ristin et al., 2014), even in the emotion recognition field (Gonzalez and Prevost, 2021).

In the era of big data, with the increase in the size of databases, the field of machine learning faces a challenge, the creation of ground truth, which can be costly in time and effort. We are therefore increasingly finding ourselves in contexts of incomplete supervision, where we are given a small amount of labeled data, which is insufficient to train a good learner, while unlabeled data is available in abundance. To this end, different learning techniques have been proposed (Zhou, 2018) with human intervention such as active learning (Settles, 2009) or without human intervention such as semi-supervised methods. One of these last ones is based on disagreement methods (Zhou and Li, 2010), co-training being one of its most famous representations.

Co-training is a learning technique proposed in 1998 by Blum and Mitchell (Blum and Mitchell, 1998) which is traditionally based on the use of two machine learning models. The main idea is that they complement each other: one helps the other to correct the mistakes it does not make, and vice versa. A second idea is to exploit data that are not labeled (present in large quantities), rather than processing only labeled data (present in small quantities). For

this to work, the dataset must be described according to two independent views, i.e., two different representation spaces for this same dataset. Otherwise, their potential to provide each other with relevant information is limited. Although, this assumption is difficult to achieve in practice, as Wang and Zhou explain in 2010 in their study on co-training (Wang and Zhou, 2010).

During co-training, estimations of probabilities are mainly needed (see Sec.5.1) and act as confidence levels. For a decision tree (DT), this is deduced from the statistics stored in the leaves during the learning process. Unfortunately, DTs are considered as poor estimators of these probabilities and suggestions have been made to compute the posterior probabilities differently, one of them being to apply a Laplace correction (Provost and Domingos, 2003)(Tanha et al., 2017). Semi-supervised learning was first proposed in 2003 to effectively use unlabeled data for facial expression recognition with the Cohn-Kanade dataset (Cohen et al., 2003). It consists in switching models during co-training when performance collapses. Indeed, one of the drawbacks of co-training is the labeling error that can occur during co-training iterations. This effect can then be tackled either by placing less confidence in the model, by setting a higher threshold, or by looking for strategies to correct these errors on the fly (Zhang et al., 2016).

An observable limitation in the co-training procedure is that the classifier retains multiple training sessions on the same set of data. We could take advantage of the progress in incremental learning to make only one pass and update the tree while pseudo-labeling the data without re-training from scratch at each iteration.

3 DATASETS AND FEATURE EXTRACTION

3.1 Datasets

3.1.1 Compound Facial Expressions of Emotion (CFEE)

The CFEE dataset contains 230 subjects with one image for each of the 22 categories present in the dataset: 6 basic emotions (anger, surprise, sad, happy, fear, disgust), 15 compound emotions (i.e. a combination of two basic emotions), and the neutral expression (Du et al., 2014). For each subject, we selected 7 images with the six basic emotions and the neutral face (the dataset doesn't have the contempt emotion). Thus, 1285 images are retained to train the NCMF baseline classifier and 322 for evaluation.

3.1.2 Extended Cohn-Kanade CK+

The CK+ dataset is the most popular database in the field of emotion recognition. It contains 327 labeled sequences of deliberate and spontaneous facial expressions from 123 subjects, 85 females and 38 males (Lucey et al., 2010). A sequence lasts approximately 20 frames in average (from 4 to about 60), and always begins with a neutral expression, then progresses to a specific expression until a peak in intensity (apex) that is labeled using the Facial Action Coding System (FACS) (Ekman, 1997). By collecting the labeled images (without including the emotion of contempt), and by focusing on neutrals and apexes we collect 1802 images labeled among the 6 basic emotions (590 for men, and 1202 for women).

3.2 Feature Extraction

The OpenFace library developed by (Baltrušaitis et al., 2016) was used to extract 68 facial landmarks and high level features, namely, facial Action Units (AUs). We then extracted Local Binary Patterns (LBP) and Histograms of oriented Gradients (HoG) features with the scikit-image library from the cropped faces registered by OpenFace.

4 NEAREST CLASS MEAN FOREST

4.1 Original Algorithm

Nearest Class Mean Forest (NCMF) (Ristin et al., 2014) is a Random Forest (RF) (Breiman, 2001) whose nodes are Nearest Class Mean (NCM) (Hastie et al., 2009) classifiers. In these nodes, two class centroids (called c_i and c_j) are computed. They are used to direct samples x (given a distance measure) to the left child of the node (if x is closer to c_i than c_j) or its right child (otherwise). A class bagging occurs during training: only a random subset of available classes is considered in each node. The splitting decision function is also modified. Among the data available in the current node n, we can compute all the possible pairs of centroids. The optimal pair is the one that maximizes the Information Gain I (Quinlan, 1986).

The main advantage of NCM forests is their ability to learn new data and classes incrementally. Two incremental strategies, namely, Update Leaf Statistics (*ULS*) and Incremental Growing Tree (*IGT*) have

been introduced in (Ristin et al., 2014). The incremental data is propagated, as in a prediction phase, in each tree of the forest; then, the occurrences at the level of the predicting leaves are updated (ULS). Since the ULS strategy only updates the distributions in the leaves of the tree, when incremental data appear, the distributions evolve and thus the predictions are likely to change. With the IGT strategy, we first proceed as with ULS; then, right after the update of the class distributions of the leaf, we check if the increment satisfies a condition that could locally lead to the construction of a subtree, e.g. majority label modification. If this is the case, the leaf is transformed into a node, which triggers the recursive construction of the subtree from this position. The data considered by the subtree are all those that were present in this leaf, either during the learning or during the incrementation.

4.2 Probabilistic Decision Criterion

Classically, for decision trees, a test sample is propagated from the root node to a terminal leaf. Then, the decision is made by the majority class present in this leaf. For tree forests, a majority vote is applied on tree decisions.

As explained above, co-training needs to evaluate the confidence we have in each view for a given sample. So, we need to compute a posterior probability vector. Then, using the maximum a posteriori rule, we take the highest probability and use a confidence threshold to decide whether x will be used for incremental training or not.

Given a sample x, let $\phi_\mu : \mathbb{R}^m \mapsto \mathbb{R}^l$ be the prediction function associated with the model μ and returning a vector containing the class conditional *posterior* probabilities:

$$\forall x \in \mathbb{R}^m, \phi_\mu(x) = [P(k_1|x), ..., P(k_l|x)], \quad (1)$$

where

- $\mathcal{K} = \{k_1, ..., k_l\}$ is the set of l labels,
- x is an observation to be classified,
- $P(k_i|x)$ the conditional (posterior) probability that x belongs to the class k_i, for $1 \leq i \leq l$.

The class assigned to the observation x, $l^{i+}(x)$, is then determined by the rule of the maximum *a posteriori*:

$$l^{i+}(x) = \underset{k_i \in \mathcal{K}}{argmax}[P(k_1|x), ..., P(k_l|x)]. \quad (2)$$

We establish a **confidence criterion**, consisting in attributing a threshold θ such that:

$$\max [P(k_1|x), ..., P(k_l|x)] \geq \theta. \quad (3)$$

If the forest is composed of t trees, we have t prediction vectors for a test sample. It is thus necessary to define an operator to combine optimally these vectors.

Consider, for an observation x and a decision tree t, the vector $S^t(x) = [S^t(k_1|x), ..., S^t(k_l|x)]$, where $S^t(k_i|x)$ corresponds to the number of occurrences of the class k_i in each leaf of t. The vector ϕ_μ is then computed as follows:

$$\phi_\mu(x) = \frac{1}{R(x)} \sum_{t \in T} S^t(x). \quad (4)$$

with,

$$R(x) = \sum_{t \in T} \sum_{i=1}^{l} S^t(k_i|x). \quad (5)$$

We first aggregate the values of all the predictor leaves and then, calculate the **global** probability at the forest level (4). Thus, a leaf with a smaller number of class members than another leaf will have less impact in the final calculation of the $\phi_\mu(x)$ probabilities.

5 CO-INCREMENTATION ALGORITHM

5.1 Original Co-Training Algorithm

Co-training (Blum and Mitchell, 1998) is a semi-supervised learning technique that can be used when a dataset is partially labeled. It involves the collaboration between two machine learning models. Let V_1 and V_2 be two families of features, also called "views", fully describing each observation of the dataset $x = (V_1(x), V_2(x))$. The corresponding datasets are denoted $L^{[V_1]}, L^{[V_2]}$ for labeled data and $U^{[V_1]}, U^{[V_2]}$ for unlabeled data. Each model is trained on a view and both models must satisfy the independence assumption.

1. ***Pre-Training***: each model initially trains on its own set labeled $L^{[V_1]}$ or $L^{[V_2]}$.

Co-training is an iterative process. For each observation of U, the following 2 steps are performed:

2. ***Labeled Set Extension***: each model predicts a pseudo-label for the observation; the most reliable prediction (given a confidence criterion) is used to add the observation and its most reliable pseudo-label to the labeled set of the other view, $L^{[V_2]}$ if model 1 was the most reliable and vice versa ;
3. ***Self-Training***: the model whose pseudo-label was the least reliable is re-trained on the new labeled set.

We can notice that single-view classifiers require a complete re-learning at each iteration. This process can become expensive in terms of computational time, especially if the size of U is large.

5.2 Incremental Co-Training Algorithm (EBSICO)

In this research work, we propose a co-training method that differs from the classical method (see Sec.5.1) by using a hybrid method, combining semi-supervised learning and incremental learning paradigms. Our aim is not to build generic classifiers but to personalize generic classifiers to a subset of subjects. Thus, we use a first dataset G to build these generic single-view classifiers. Then, the second dataset I is used to personalize these classifiers incrementally by using a co-training based algorithm. The main advantage of this process is to avoid re-training from scratch the single-view classifiers through co-training iterations. Fig. 1 shows the different steps of our method that are described below:

1. ***Pre-Training***: At step 1, generic single-view classifiers are trained respectively on their own views $G^{[V_1]}$ and $G^{[V_2]}$. These are the reference models, and will be referred to by the name of their view.
2. ***Error-Based Self-Incrementation (EBSI)***: At step 2, the model associated to the view V_i predicts a class for each of the observations of $L^{[V_i]}$. If this predicted class is different from the ground truth (the dataset $L^{[V_i]}$ is labeled), the model increments on this example, with a $INCR()$ function. This error-based incremental strategy is possible since we work on generic models that are already trained. This step is described in the algorithm 1.
3. ***Error-Based CO-incrementation (EBCO)***: For each **unlabeled** x observation of $U^{[V_i]}$,

(A) We identify the most reliable model, using the posterior probabilities $p^{1+}(x)$ and $p^{2+}(x)$:

$$p^{1+}(x) = \max \phi_{\mu_1}(x). \quad (6)$$

$$p^{2+}(x) = \max \phi_{\mu_2}(x). \quad (7)$$

The largest posterior probability thus informs us about the most reliable model for predicting the pseudo-label of x. We use the predictions of the models $l^{1+}(x)$ and $l^{2+}(x)$ (see Eq.2) as the *pseudo-label*.

(B) Then, the less reliable model is incremented in a supervised manner from the unlabeled observation using the pseudo-label provided by the more reliable model. The algorithm 2 describes more precisely the co-incrementation procedure.

Contrary to the classical co-training algorithm, with our approach *EBCO*, for each iteration, when the maximum probability of belonging to a class exceeds the confidence threshold *EBCO*, the least reliable model does not restart its learning from the beginning. It only increments on the observation of the considered iteration. Moreover, it increments only if its pseudo-label differs from the one delivered by the most reliable model.

Algorithm 1: Error-based Self-incrementation (EBSI).

Require: generic model μ_i pretrained on $G^{[V_i]}$
  for all $(x,y) \in L^{[V_i]}$ **do**
    The model uses the function $\phi(x)$ to obtain the probability vector of class membership:
    $p^{(i)} \leftarrow \phi(x^{[V_i]})$
    The predicted label is thereby determined as:
    $l^{i+} \leftarrow argmax(p^{(i)})$
    if $l^{i+} \neq y$ **then**
      $\mu_i \leftarrow INCR(\mu_i, x, y)$
    end if
  end for

Algorithm 2: Error-Based Co-Incrementation (EBCO).

Require: models μ_1 and μ_2 incremented following **EBSI** method, $\theta \geq 0$
  for all $u \in U$ **do**
    Each model uses the function $\phi_\mu(u)$ to obtain the probability vectors of class membership:
    $p^{(1)} \leftarrow \phi_{\mu_1}(u^{[V1]})$
    $p^{(2)} \leftarrow \phi_{\mu_2}(u^{[V2]})$
    The pseudo-labels are then:
    $l^{1+} \leftarrow argmax(p^{(1)})$
    $l^{2+} \leftarrow argmax(p^{(2)})$
    The probabilities associated are then:
    $p^{1+} \leftarrow max(p^{(1)})$
    $p^{2+} \leftarrow max(p^{(2)})$
    if $p^{1+} > p^{2+}$ **and** $p^{1+} \geq \theta$ **then**
      if $l^{2+} \neq l^{1+}$ **then**
        $\mu_2 \leftarrow INCR(\mu_2, u^{[V2]}, l^{1+})$
      end if
    else if $p^{2+} > p^{1+}$ **and** $p^{2+} \geq \theta$ **then**
      if $l^{1+} \neq l^{2+}$ **then**
        $\mu_1 \leftarrow INCR(\mu_1, u^{[V1]}, l^{2+})$
      end if
    end if
  end for

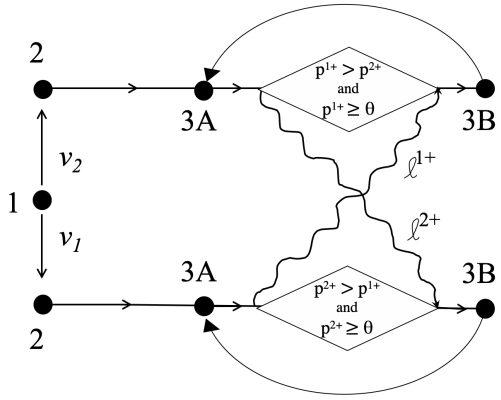

Figure 1: Diagram describing the entire proposed EBSICO process - (1): pre-training on the first dataset, (2): incrementing on labeled samples from the second dataset (EBSI), (3) incrementing on unlabeled samples from the second dataset (EBCO), using for each sample the prediction of the most reliable model as a pseudo-label (A) and incrementing the least reliable model with this pseudo labeled sample (B).

Figure 2: Example of a video sequence from CK+ for the surprise.

6 EXPERIMENTAL PIPELINE AND RESULTS

6.1 Data Preparation

To carry out our experiments, in a first stage we performed data partitioning. The CFEE dataset is used to train the generic single view classifiers. Thus, it corresponds to the dataset G. The CK+ database was divided into two subsets named I and E, used respectively for incremental learning and evaluation. Next, we split I into L and U subsets corresponding to the labeled and unlabeled sets, such that $L \cup U = I$. For each sequence of n images (e.g. see Figure 2), we noted i_0 and i_1 the first and second image corresponding to the neutral emotion and i_{n-1} and i_n the two last images of the sequence, corresponding to the maximum intensity emotion (**apex**). We aim to evaluate the ability of co-incrementation to recognize forced expression after learning subtle ones. For this purpose, the images were assigned to the subsets E and I as follows:

- E contains the set of images i_0 and i_n
- I contains the set of images i_1 and i_{n-1}

6.2 Reference Model Training

As described in section 3.2, we extracted low-level texture features and high-level facial Action Units. So, each sample x is described by two views: $x = (AU, TX)$ where AU corresponds to the Action Units vector and TX to the concatenation of LBP and HoG vectors.

The NCMFs models were first trained in a supervised manner on fully labeled CFEE using AU and TX views. These models, corresponding to step 1 of the EBSICO procedure, are generic ones. They will be used as reference models regardless of the personalization strategy. For each clustering criterion, we refer to these models respectively by the names $AU^{[\Theta]}$ and $TX^{[\Theta]}$ (see below).

6.3 Personalization Chunking

The purpose of the second stage is to apply data personalization, which consists in performing data clustering. We are going to separate the data into slots by grouping them, according to a criterion called *personalization*. For each of these criteria where Θ is the acronym of personalization, we name these models respectively by $AU^{[\Theta]}$ and $TX^{[\Theta]}$. The proposed customizations are as follow:

ALL: No customization is an approach that serves as a baseline and consists of applying the co-incrementation algorithm directly on all the data in I, as one would do for a generic model. It uses only one group containing all the observations of I. In the following, we refer to this strategy as *ALL*.

GENDER: The gender personalization approach performs a clustering of the data according to the perceived gender. We obtain two groups of data named M (*man*) and W (*woman*) corresponding to men and women data respectively. The gender information has been assigned manually. In the following, we refer to this strategy as *GENDER*.

MORPHO: The morphological personalization approach performs a clustering of the data according to the face morphology. To do so, we used the landmarks (see Sec.3.2) detected on the neutral face of each subject. Then, the K-means algorithm was used on these data to identify several clusters based on morphological features.

We decided to divide the subjects of the dataset into 8 slots which we considered as a good compromise in terms of number of images per slot and number of models; figure 3 illustrates the variation of the total intra-class variance for a number k of clusters and confirms us in this choice. In the following, we refer to this strategy as *MORPHO*.

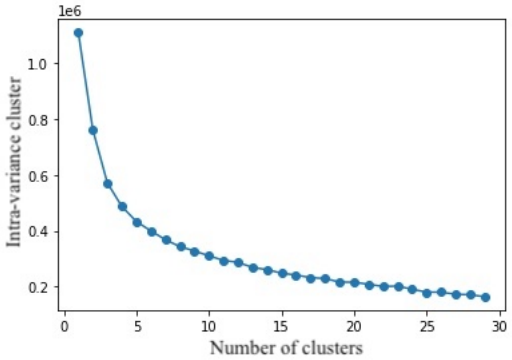

Figure 3: Evolution of the intra-class variance according to the number of clusters (*elbow plot*).

Note that each group (*cluster*) is said to be *subject-independent*. On the one hand, these data have not been seen during the training of the generic models. On the other hand, all images of the same subject belong to the same cluster and the same cluster can contain data of several subjects. For a given cluster, the associated sets I and E contain the same individuals but different images (in terms of emotional intensity), in order to evaluate the impact of personalization on the reference models after co-incrementation.

6.4 Chunk Specialization

In the third stage of the proposed pipeline, the EBSICO process is executed within each chunk as follows. In our experiments we considered a θ value of 0.8 and used NCM forests of 50 trees with the *IGT* strategy as the incrementation *INCR*() function (Ristin et al., 2014).

We decide to use a ratio of 5% of labeled data, so, in the following, $|L| = 0,05 * |I|$ and $|U| = 0,95 * |I|$.

Then, we execute sequentially and for each single-view classifiers:

- *EBSI* on L
- *EBCO* on U

6.5 Model Evaluation per Chunk

The last step of the pipeline consists in evaluating each model per chunk at different stages of the EBSICO procedure, in order to follow its evolution in terms of personalization. Thus, per view, and per chunk, we obtain a performance score that we measure when the model is in the baseline state, at the end of the EBSI procedure, and at the end of the EBCO procedure.

To carry out our experiments, we used the accuracy metric to evaluate the model performance individually:

$$acc = \frac{\sum_{x \in X} I(y = l^{i+}(x))}{|Y|} \tag{8}$$

where y is the class (ground-truth) of x, the expression $\sum_{x \in X} I(y = l^{i+}(x))$ corresponds to the number of correct predictions and $|Y|$ to the total number of observations to be labeled.

Likewise, to evaluate the contribution of a specific personalization criterion Θ which may contain several chunks, we used:

$$acc^{[\Theta]} = \frac{\sum_{c \in C} acc(c) \times |c|}{|C|} \tag{9}$$

where C represents the set of chunks and c corresponds to the samples belonging to a chunk.

6.6 Experimental Results

6.6.1 Impact of Co-Incrementation

The purpose of this analysis is to evaluate the contribution provided by the whole co-incrementation process (*EBSICO*). Tables 1 and 2 describe the results obtained for the single-view models trained respectively on the *AU* and *TX* views. The first column reports the baseline accuracy (before co-incrementation). The second and third columns report the accuracy after applying sequentially *EBSI* and *EBCO* processes.

We can observe that for low labeling rates, the *EBSI* process does not influence and in some cases decreased the prediction performance. Thus, performing self incremental learning with only 5% of labeled data is not sufficient to improve the model performance.

On the other hand, we observe that the *EBCO* process improves the performance compared to the baseline and the *EBSI* process regardless of the personalization method chosen. Hence, co-incrementation is robust enough regardless of the little amount of labeled data.

For a deeper analysis, we also studied the evolution of the disagreement measure proposed by (Shipp and Kuncheva, 2002). This is the rate of non-common errors when one model makes the right prediction and not the other. In other words, it is a measure

of conflict between both models. The results obtained by the model $(AU^{[ALL]}, TX^{[ALL]})$, according to different labeled data rates, are shown in Fig. 4. First, we can observe that the disagreement measure for the baseline models is 0.2. In other words, before co-incrementation, our models satisfy quite well the independence assumption, necessary condition for the co-training to work. The figure also shows that the *EBCO* procedure drastically reduced the non-common error rate converging almost to 0. Thus, co-incrementation also helped to reduce the common error rate between models. Consequently, both models were able to improve their performance and the process helped the best model to increase its initial accuracy.

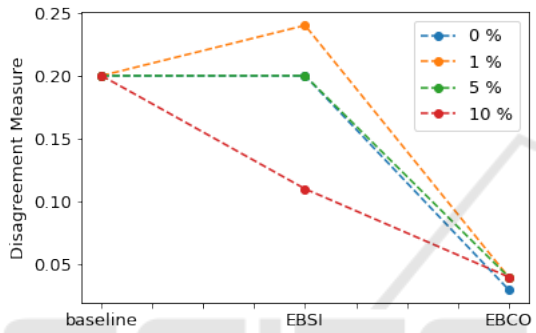

Figure 4: Evolution of the non-common error rate through the different stages of the EBSICO framework, according to the rate of labeled data.

Table 1: Accuracy measures at different stages of the EBSICO framework with **5%** of labeled samples - Action Units.

Model	*baseline*	*EBSI*	*EBCO*
$AU^{[ALL]}$	0.946	0.947	0.948
$AU^{[M]}$	0.946	0.949	**0.953**
$AU^{[W]}$	0.946	0.939	0.947
$AU^{[C1]}$	0.950	0.950	**0.962**
$AU^{[C2]}$	0.931	**0.954**	0.931
$AU^{[C3]}$	**0.974**	**0.974**	0.962
$AU^{[C4]}$	0.875	0.917	**0.958**
$AU^{[C5]}$	0.950	0.950	**0.966**
$AU^{[C6]}$	0.933	0.933	**0.958**
$AU^{[C7]}$	**0.967**	0.963	0.949
$AU^{[C8]}$	0.911	0.931	**0.960**

Table 2: Accuracy measures at different stages of the EBSICO framework with **5%** of labeled samples - Textures.

Model	*baseline*	*EBSI*	*EBCO*
$TX^{[ALL]}$	0.811	0.866	**0.943**
$TX^{[M]}$	0.766	0.847	**0.939**
$TX^{[W]}$	0.833	0.894	**0.936**
$TX^{[C1]}$	0.843	0.824	**0.931**
$TX^{[C2]}$	0.713	0.701	**0.885**
$TX^{[C3]}$	0.821	0.833	**0.974**
$TX^{[C4]}$	0.625	0.625	**0.958**
$TX^{[C5]}$	0.891	0.950	0.950
$TX^{[C6]}$	0.849	0.840	**0.941**
$TX^{[C7]}$	0.808	0.869	**0.944**
$TX^{[C8]}$	0.752	0.832	**0.921**

Moreover, our models were overall able to improve their performance on a new distribution of data by using only a small amount of labeled data. This is typically a problem of domain adaptation (Csurka, 2017) where distributions of the training and test sets do not match, as we have here with datasets CFEE and CK+. In such a configuration the performance at test time can be significantly degraded. However, this problem is beyond the scope of this paper.

6.6.2 Impact of Personalization

The purpose of this analysis is to evaluate the contribution provided by the personalization solution we have proposed.

Results showed that $AU^{[MORPHO]}$ obtained the highest accuracy rate compared to $AU^{[GENDER]}$ and $AU^{[ALL]}$ (see Table 3). This can be due to the fact that clustering images according to the morphological criterion allowed subjects with common characteristics to be optimally isolated into groups. Thus, it allowed each model to specialize on a group of subjects with common traits. Indeed, landmark position showed itself to be a more robust criterion for data separation than gender criterion.

On the other hand, comparing $AU^{[ALL]}$ with $AU^{[GENDER]}$ and $AU^{[MORPHO]}$, $AU^{[ALL]}$ presented the lowest performance rate, for 5% and 10% of labeled samples, and the second lowest for 0% and 1%. This leads to the conclusion that the customization process increased the robustness of the model.

For further analysis, we also computed the mean samples numbers per model. $AU^{[ALL]}$ contained the

largest amount of labeled data since it did not use data clustering. Considering a label rate of 1%, 5% and 10%, we computed an average of 4, 22 and 45 labeled samples per chunk for $AU^{[GENDER]}$ and an average of 1, 5 and 11 labeled samples per chunk for $AU^{[MORPHO]}$ (see Table 4). Regardless of the little amount of labeled data per chunk, $AU^{[MORPHO]}$ was capable of obtaining better performance compared to $AU^{[GENDER]}$ and $AU^{[ALL]}$. As a consequence, we can deduce that a rationalized clustering strategy provides robustness during the co-incrementation process, regardless of the number of labeled samples. Therefore, a rationalized way for the data clustering process is crucial for models improvements.

The table also shows that thanks to personalization, we allow the labelling of fewer images on average than the generalized $AU^{[ALL]}$ model.

Furthermore in Table 1, it was observed that both AU and TX benefited from co-incrementation. This confirms our hypothesis that knowledge sharing between different models leads to improvement of the prediction rate, regardless of the view and the chosen clustering criterion.

When several subjects produce the same emotion differently, as a direct consequence of the identity bias, this results in a greater intra-class variance. We observed the evolution of this variance, according to the separation of the slots according to the personalization criteria that we have presented in the previous section. The number of slots created depends on the chosen personalization criteria. In order to make a fair comparison, we have therefore carried out for each criterion, except for *ALL*, a random separation with the same number of slots. This has been carried out over 100 folds, and the final result is the average of the slot intra-class variances with their standard deviations. In this way, we distinguish a separation made on a random criterion from a separation made on a rational criterion. Moreover, because of the great number of neutrals, this one was *downsampled* to 50 to correspond to the distributions of the other labels. The separation criteria with their associated number of slots are the following:

Table 3: average accuracy per chunk - EBSICO with Action Unit view.

%	$AU^{[ALL]}$	$AU^{[GENDER]}$	$AU^{[MORPHO]}$
0	0.947	0.946 ± 0.0	**0.952** $\pm$ 0.013
1	0.948	0.946 ± 0.0	**0.952** $\pm$ 0.013
5	0.948	0.949 ± 0.003	**0.956** $\pm$ 0.01
10	0.951	0.954 ± 0.001	**0.956** $\pm$ 0.01

Table 4: Chunk sizes - EBSICO - Action Unit view.

%	$\lvert L^{ALL}\rvert$	$\lvert L^{GENDER}\rvert$	$\lvert L^{MORPHO}\rvert$
0	0	0	0
1	9	4 ± 2	**1.0** $\pm$1
5	45	22 ± 8	**5** $\pm$3
10	90	45$\pm$15	**11** $\pm$ 6

Table 5: Impact of chunking on intra-class variances.

ALL	4.719
RANDOM 2	4.623 ± 0.167
GENDER 2	4.466 ± 0.624
RANDOM 5	4.327 ± 0.353
MORPHO 5	4.211 ± 0.304
RANDOM 8	4.035 ± 0.488
MORPHO 8	3.935 ± 0.483

ALL: 1 slot (whole CK+incr, AUs only),

RANDOM 2: 2 slots, random separation, (average over 100 folds),

GENDER 2: 2 slots,

RANDOM 5: 5 slots, random separation, (average over 100 folds),

MORPHO 5: 5 slots,

RANDOM 8: 8 slots, random separation, (average over 100 folds),

MORPHO 8: 8 slots.

The results are available in the table 5. We can notice that, the more slots there are, the more the intra-class variance decreases. In an extreme case like putting only one subject per slot, the intra-class variance will tend towards 0. We can thus observe in the table that the average intra-class variance decreases when we constitute more slots with a different criterion. These results suggest that as the intra-class variance decreases, the identity bias is reduced. Moreover, when comparing the same number of slots, we observe that the rational separation criterion offers a slightly lower intra-class variance than the random criterion. This result confirms, on the one hand, the quality of these personalization criteria, and on the other hand, motivates the search for even more refined rational separation criteria for future experiments.

Table 6: Comparison of the classical co-training algorithm with our co-incrementation algorithm.

Methods \Metrics	acc after L (90)	acc after U (811)	execution time
classical co-training	$(0.957, 0.888)$	$(0.91, 0.909)$	245 min
co-incrementation	**(0.96,0.897)**	**(0.954, 0.933)**	**3 min**

6.6.3 Comparison of Co-Training and Co-Incrementation

Finally, we compared our EBSICO algorithm with the classical co-training algorithm. We used the following learning process:

1. training of μ_1 and μ_2 respectively on $CFEE_A^{[AU]}$ and $CFEE_A^{[TX]}$,
2. the incrementation of the models is done from $CK+_I$ which has been divided to 10% of labels, so that the numbers are: 90 data for L and 811 data for U, without slots,
3. first evaluation of the models after incrementing on L,
4. then co-train incrementing, with the classical method (re-training from zero at each iteration) and our co-incrementing method,
5. second evaluation on $CK+_E$ after incrementing on U.

The accuracies are given in pairs: the first one corresponds to the AU model, and the second to the TX model. The confidence threshold has been set at 0.8. Finally, we compute the total execution time of the sequence, in order to compare the classical method and the incremental method.

The results, comparing the classical co-training procedure and the one we propose, are presented in the table 6. We can observe that when a significant number of data in U is present, the classical model sees its performances decrease drastically, while the incremental model stagnates, or even improves the μ_2 model. Finally, the EBSICO procedure that we propose has a major interest in terms of speed of execution, nearly 80 times faster in this experiment.

7 CONCLUSION

In this paper, we propose a hybrid method, which combines two algorithms, namely co-training and incremental learning, allowing two models to collaborate and share their knowledge. Compared to the classical co-training method that performs re-training from scratch, our approach performs model incrementation continuously on new samples, saving significant execution time. Another advantage of using incremental learning over re-training a model from scratch, in addition to the execution time, is to avoid "catastrophic forgetting". NCMFs provide robust resistance to models trained several steps earlier.

Second, in this paper we provide an in-depth analysis of model personalization for emotion recognition. Models taking into account morphological features have shown better performance versus clustering by gender. Indeed, a rationalized technique for feature clustering is crucial for co-training model performance.

Finally, our third contribution concerns the field of semi-supervised learning, more specifically, on the ability of models to increase their performance with only 5% of labeled samples, as demonstrated in our experiments. Our experiments have been conducted with small datasets, but we could imagine in future research work using this technique with larger databases.

REFERENCES

Baltrušaitis, T., Robinson, P., and Morency, L.-P. (2016). Openface: an open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10. IEEE.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32.

Chu, W.-S., De la Torre, F., and Cohn, J. F. (2013). Selective transfer machine for personalized facial action unit detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3515–3522.

Cohen, I., Sebe, N., Cozman, F. G., and Huang, T. S. (2003). Semi-supervised learning for facial expression recognition. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 17–22.

Csurka, G. (2017). Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*.

Denil, M., Matheson, D., and Freitas, N. (2013). Consistency of online random forests. In *International conference on machine learning*, page 1256–1264. PMLR.

Du, S., Tao, Y., and Martinez, A. M. (2014). Compound facial expressions of emotion. *Proceedings of the National Academy of Sciences*, 111(15):E1454–E1462.

Ekman, R. (1997). *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA.

Gonzalez, J. and Prevost, L. (2021). Personalizing emotion recognition using incremental random forests. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 781–785. IEEE.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Hu, C., Chen, Y., Hu, L., and Peng, X. (2018). A novel random forests based class incremental learning method for activity recognition. *Pattern Recognition*, 78:277–290.

Kollia, V. (2016). Personalization effect on emotion recognition from physiological data: An investigation of performance on different setups and classifiers. *arXiv preprint arXiv:1607.05832*.

Lakshminarayanan, B., Roy, D. M., and Teh, Y. W. (2014). Mondrian forests: Efficient online random forests. *Advances in neural information processing systems*, 27:3140–3148.

Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., and Matthews, I. (2010). The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 ieee computer society conference on computer vision and pattern recognition-workshops*, pages 94–101. IEEE.

Meng, Z., Gaur, Y., Li, J., and Gong, Y. (2019). Speaker adaptation for attention-based end-to-end speech recognition. *arXiv preprint arXiv:1911.03762*.

Oudot, L., Prevost, L., Moises, A., and Milgram, M. (2004). Self-supervised writer adaptation using perceptive concepts: Application to on-line text recognition. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 598–601. IEEE.

Provost, F. and Domingos, P. (2003). Tree induction for probability-based ranking. *Machine learning*, 52(3):199–215.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Ristin, M., Guillaumin, M., Gall, J., and Van Gool, L. (2014). Incremental learning of ncm forests for large-scale image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, page 3654–3661.

Sariyanidi, E., Gunes, H., and Cavallaro, A. (2014). Automatic analysis of facial affect: A survey of registration, representation, and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(6):1113–1133.

Senechal, T., Bailly, K., and Prevost, L. (2010). Automatic facial action detection using histogram variation between emotional states. In *2010 20th International Conference on Pattern Recognition*, pages 3752–3755. IEEE.

Settles, B. (2009). Active learning literature survey.

Shipp, C. A. and Kuncheva, L. I. (2002). Relationships between combination methods and measures of diversity in combining classifiers. *Information fusion*, 3(2):135–148.

Tanha, J., van Someren, M., and Afsarmanesh, H. (2017). Semi-supervised self-training for decision tree classifiers. *International Journal of Machine Learning and Cybernetics*, 8(1):355–370.

Wang, W. and Zhou, Z.-H. (2010). A new analysis of co-training. In *ICML*.

Yang, S. and Bhanu, B. (2011). Facial expression recognition using emotion avatar image. In *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pages 866–871. IEEE.

Zhang, Z., Ringeval, F., Dong, B., Coutinho, E., Marchi, E., and Schüller, B. (2016). Enhanced semi-supervised learning for multimodal emotion recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5185–5189. IEEE.

Zhou, Z.-H. (2018). A brief introduction to weakly supervised learning. *National science review*, 5(1):44–53.

Zhou, Z.-H. and Li, M. (2010). Semi-supervised learning by disagreement. *Knowledge and Information Systems*, 24(3):415–439.