Detecting Morphing Attacks via Continual Incremental Training

Lorenzo Pellegrini, Guido Borghi, Annalisa Franco, Davide Maltoni
Dipartimento di Informatica - Scienza e Ingegneria
University of Bologna, 47521 Cesena, Italy
{l.pellegrini, name.surname}@unibo.it

Abstract

Scenarios in which restrictions in data transfer and storage limit the possibility to compose a single dataset – also exploiting different data sources – to perform a batch-based training procedure, make the development of robust models particularly challenging. We hypothesize that the recent Continual Learning (CL) paradigm may represent an effective solution to enable incremental training, even through multiple sites. Indeed, a basic assumption of CL is that once a model has been trained, old data can no longer be used in successive training iterations and in principle can be deleted. Therefore, in this paper, we investigate the performance of different Continual Learning methods in this scenario, simulating a learning model that is updated every time a new chunk of data, even of variable size, is available. Experimental results reveal that a particular CL method, namely Learning without Forgetting (LwF), is one of the best-performing algorithms. Then, we investigate its usage and parametrization in Morphing Attack Detection and Object Classification tasks, specifically with respect to the amount of new training data that became available.

1. Introduction

In this paper, we address the scenario in which new sets of biometric training data become progressively available across time, even on different sites [29]. Differently from the traditional Machine Learning setting, the batch-based training procedure [2] is unfeasible, making challenging the learning process. Therefore, we investigate the use of the Continual Learning (CL) [34] paradigm to train a model in a distributed setting, in which several distinct data chunks containing personal information cannot be stored and then are available only in a limited time frame. In other words, we aim to address the problem of incrementally training a model on multiple data sources that, for different reasons (*e.g*. privacy issues), cannot be shared and stored for long time ranges, thus making it impossible to create a single training dataset, as represented in Figure 1.

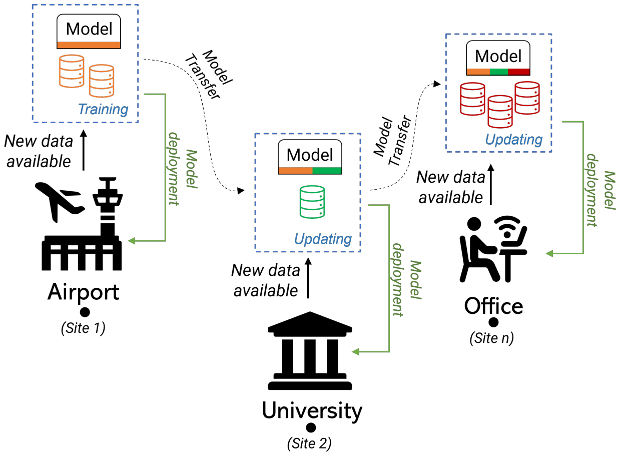


Figure 1. Visualization of the proposed incremental training scenario, in which a trained model is updated every time new chunks of data are temporarily available, even on different sites. In this manner, no data transfer is involved and Continual Learning algorithms are eligible to fully exploit the data available through incremental model training.

A practical case is represented by the development of solutions to contrast the *Morphing Attack* [14] (see Fig. 2), in which severe privacy issues strongly limit the possibility of storing, transferring and sharing public datasets of sensitive data (*e.g*. facial images, sex, and age). As a consequence, each research laboratory or institution usually exploits for training only its own data, thus developing a model with limited generalization capability whose performance is generally unsatisfactory on new unseen data [3].

From a theoretical point of view, this distributed training setting can be tackled through Federated Learning [23], which is indeed a paradigm focused on training a global model exploiting multiple clients that have access to private and not-shareable data. We observe that this recent paradigm is a promising solution, even though it presents technical challenges [22], including the development of an infrastructure that supports repeated global model transfers and that maintains a copy of the original dataset in case new training from scratch or fine-tuning procedures of the original model become necessary. Moreover, privacy issues can limit the temporal range in which new data are stored on a site, hampering the possibility to have different datasets available on different sites. For example an airport gate could store some face images only for the short time required to update a model. Finally, a limited latency between clients, that have to be simultaneously online, despite the relevant size of data transfers, is needed [32]. These issues lead us to explore a complementary approach based on the recent and interesting Continual Learning paradigm.

In particular, this paper represents one of the first attempts to investigate the use of this paradigm in the aforementioned scenario, that imposes three challenging and novel peculiarities:

- **Variable chunk size**: data amounts available at each training step (from now referred to as *experience*) are not known in advance. In other words, it is possible to update the model only through a variable amount of training samples. In our *Morphing Attack Detection* (MAD) [40] scenario, the model is kept updated, for instance, every time a certain (and variable) amount of new data becomes available (*e.g.* a laboratory has collected a new dataset, new morphed images are generated with new algorithms or additional bona fide images are collected through an Automatic Border Control system in an international airport, etc.)

- **Variable amount of training steps**: the number of learning experiences is not known in advance. It is unpredictable to define how many times a model receives new data to update the knowledge. It follows that we aim to obtain the best performance after each training phase of the model, in order to improve or, at least, fully preserve the model performance in the MAD task. Therefore, a metric able to consider not only the final performance but the accuracy across the whole learning process, referred to as BRoT, is introduced in this paper, as detailed in Section 4.3.1.

- **Limits in storage**: limitations in the release and transfer of datasets inhibit the direct use of stored samples. However, we observe that in the CL paradigm, the use of these samples, referred as replay memory [6], is one of the most effective ways to contrast the so-called *catastrophic forgetting* problem [31], *i.e.* the tendency of a model to abruptly and drastically forget the previously learned knowledge. A workaround may consist in exploiting a replay memory based on embeddings instead of real samples [36], but further investigations related to privacy constraints are still needed and are out of the scope of this paper.

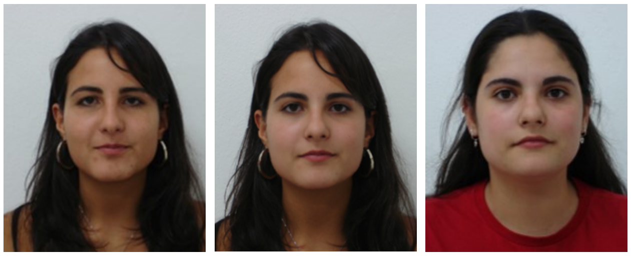

(a) Accomplice (b) Morphed (c) Criminal

Figure 2. Example of the Morphing Attack [14], in which a morphed image is obtained merging the identities of an accomplice and a criminal. The resulting face can fool automatic and human face verification-based controls.

We observe that, even in the literature related to Continual Learning, these aspects are not yet fully investigated. Therefore, in this paper, we simulate the proposed incremental training scenario addressing the MAD task. In addition, the validity of our findings is further assessed on the object classification task. We test and compare different traditional Continual Learning approaches, focusing in particular on the Learning without Forgetting (LwF) [25] method.

Experimental results reveal that the proposed setting represents an interesting and challenging scenario for common CL methods. Moreover, the choice of the proper parametrization of LwF with variable experience size is not trivial and must be investigated.

2. Morphing Attack Detection

In recent years, several studies [40, 45] confirmed that existing Face Recognition Systems (FRSs) or, more in general, face verification-based algorithms [5], are highly sensitive to specific kinds of manipulation and, in particular, to the morphing process. This vulnerability increases the probability of success of a possible morphing attack, which consists in merging two facial identities (a *criminal* and an *accomplice*) into a single-face morphed image (see Fig. 2), creating a new hybrid identity and thus destroying the link between the document and its real owner. As a consequence, it is possible to deceive the authority into issuing a document that contains the morphed image. This creates a situation where two individuals can share the same legal document, such as the electronic Machine Readable Travel Document (eMRTD). Once the morphed document is in possession, it can be used to deceive both human officers and automatic face recognition-based controls commonly used, for instance, in airports [44].

Due to the importance of this type of attack, MAD methods are strongly needed by private and public institutions. Unfortunately, these methods usually suffer from limited generalization capabilities mainly due to the lack of public datasets, which also hamper the reproducibility of the training procedure.

These limitations are amplified with MAD models based on deep learning architectures, that are prone to overfit on small low-varied datasets [3].

In this paper, we focus on the specific task of *Differential Morphing Attack Detection* (D-MAD) systems, *i.e.* methods that receive a pair of images as input [4]. In particular, the first image is the one stored in the document (*i.e.* suspected morph) while the second is a trusted live captured image. These methods work under the assumption that it is possible to compare the two input images (one of which is surely genuine) to detect the presence of the morphing attack. Generally, this approach achieves better performance with respect to MAD systems that receive as input only a single image (S-MAD). From a general point of view, our MAD task can be considered a binary classification task, with the two classes "morphed" and "bona fide".

3. Incremental MAD

In this Section, we define the tasks involved in the development of MAD systems incrementally trained. In particular, we formulate the terminology used in the rest of the paper and, for the sake of readability, we briefly recall the Continual Learning paradigm.

3.1. Incremental Training

Following [3], we formally define two key elements of the proposed scenario:

- **Learning Experience** (l): the given model M is trained on a specific chunk of data of variable size. Then, a learning experience is defined as:

  $$l_i = (M_k,\, d_i),\, 1 \leq i, k \leq N \tag{1}$$

  where M_k is the model trained at the k-th experience and updated using a new set of data $d_i \in D$, where $D = \{d_i,\, i = 1, .., N\}$ is the entire set of training data available and $N = |D|$ is the total number of data chunks accessible for the training experiences.

- **Testing experience** (t): the given model M is tested after each learning experience on the same set of testing datasets, in order to globally monitor the model performance. Formally:

  $$t_i = (M_k,\, E),\, 1 \leq i, k \leq N \tag{2}$$

  where M_k is the model updated at the k-th learning experience, E is the set of the testing datasets and $N = |D|$ is the total number of datasets as before. We observe that the size, the order, and the amount of data chunks are irrelevant to the testing procedure since no training steps are performed. E is a fixed set in order to compute comparable performance metrics after a given training experience.

Therefore, the proposed incremental scenario is formally described as:

$$B = (l_i,\, t_i),\, i = 1, ..., N \tag{3}$$

or rather as an ordered set of training experiences $l_i \in L$ computed on a specific chunk of data of variable size, each of them followed by a testing experience $t_i \in T$ used to monitor the model performance across time. Since the single chunks of data d_i are not shareable, model M is transferred each time to be updated through Continual Learning techniques to contrast the catastrophic forgetting [31].

3.2. Continual Learning

Continual Learning, also known as lifelong learning, is the ability to continually acquire, fine-tune and transfer knowledge across time [34]. This is an ability naturally present in humans and animals, but not in artificial learning systems, especially if based on deep learning architectures. In particular, computational systems, that commonly are trained on stationary batches of training data, have difficulties in acquiring new incremental knowledge from non-stationary data distributions due to the catastrophic forgetting problem [31].

The Continual Learning paradigm greatly differs from the Machine Learning one, in which the development of a learning agent is divided into two distinct phases: learning and deployment. Indeed, training data, collected only before the learning phase, are unrealistically supposed to be representative of all the nuances of future test data [16].

A variety of approaches have been proposed in the literature to limit or contrast the forgetting, ranging from regularization methods [1], that exploit constraints on the update of the neural weights, to dynamic architectures [13], in which changes in architectures are introduced to deal with the new information, and memory replay methods [48], based on the storing of past data used for current training procedures. A further analysis of the CL method investigated in this paper is reported in Section 5.1.

4. Experiments

4.1. Datasets

Idiap Morph [41, 42] collects images belonging to different datasets, *i.e.* FRGC [37], Feret [38] and Face Research Lab London Set (FRLL) [8]. Morphed images are produced through 5 different morphing algorithms, *i.e.* OpenCV [43], FaceMorpher [39], AMSL [33], StyleGAN [20] and WebMorph [8]. The quality of resulting morphed images is usually medium-low since artifacts commonly produced by landmark-based morphing algorithms or GAN generation are visible in the majority of images. No manual or automated retouching is applied.

Progressive Morphing Database [15] (PMDB) consists of more than 1000 morphed images produced using the algorithm described in [15], and accomplices and criminals are selected in AR [30], FRGC [37], and Color Feret [38] datasets. In total, 280 subjects (134 males and 146 females) are available. In morphed images are visible some artifacts, such as ghosts and blurred areas, especially close to the nose, eyes, and mouth.

MorphDB [15] dataset is composed of 100 high-quality morphed images, from an equal number of male and female subjects (50). This is one of the few datasets in which images have been manually retouched to hide artefacts produced during the morph operation and then is effective to test the performance of MAD systems. This dataset is not publicly available, but the FVC-onGoing [12] platform offers the possibility to test it as a sequestered dataset.

4.2. Configuration

Inspired by the state-of-the-art method [46] in the D-MAD scenario, the model M is a Multi-Layer Perceptron (MLP) that processes extracted features. This MLP receives as input features extracted through the "ArcFace" [11] network trained on the merge of VGGFace2 [7], CASIA [50], and MS1MV2 [17] datasets[1].

In all experiments, MLP has the same architecture that consists of 5 layers that have $512, 250, 125, 64, 2$ neurons, respectively. The activation function is ReLU, while the loss function is the Categorical Cross Entropy (CCE). As for the optimizer, we use SGD with a learning rate of 10^{-2} and a momentum of 0.9. No weight decay is applied.

The training set D consists of (morphing algorithm - dataset): StyleGAN - Feret; OpenCV - FRGC; FaceMorpher - FRLL. The whole MorphDB dataset is used for the test, consisting of pairs with both the criminal and the accomplice. In this manner, we perform a challenging cross-dataset evaluation, limiting the influence of overfitting on the investigated algorithms. Since $|D| = 4$, we permute all possible training dataset orders ($4! = 24$ orders in total) for MAD experiments. All the training datasets are split into chunks $d_i \in D$ of variable size, depending on the experimental validation conducted and detailed in the following.

4.3. Metrics

MAD task is evaluated through metrics commonly used in the literature [40]. The Bona Fide Presentation Classification Error Rate (BPCER) represents the proportion of bona fide images wrongly classified as morphed:

$$\text{BPCER}(\tau) = \frac{1}{N}\sum_{i=1}^{N} H(b_i - \tau) \tag{4}$$

Attack Presentation Classification Error Rate (APCER) represents the proportion of morphed images wrongly accepted as bona fide:

$$\text{APCER}(\tau) = 1 - \left[\frac{1}{M}\sum_{i=1}^{M} H(m_i - \tau)\right] \tag{5}$$

In both definitions, τ is the score threshold on which b_i, m_i, the detection scores, are compared; $H(x) = 1$ if $x > 0, 0$ otherwise is defined as a step function. Being error rates, low values are desired.

To summarize metrics across different testing experiences t_i, we compute the Area Under the Curve (AUC) metric. AUC is similar to the Average Mean Class Accuracy (AMCA) metric [16], but it is obtained through the trapezoidal rule and the final value is divided by the number of training experiences. In the MAD task, the AUC is computed by adding the EER, the error rate for which both BPCER and APCER metrics are equal, and the lowest point in which BPCER with APCER $\leq 1\%$ (typical working point of face verification-based systems).

4.3.1 BRoT Metric

Finally, following the aforementioned considerations about the measurement of performance during the whole learning procedure, we introduce an additional metric named *Borda Ranking over Time* (BRoT), computed over a set of algorithms $\mathcal{A}$, based on the idea of rewarding the algorithms that perform better at each testing experience. Let $r(a_j, t_i)$ be the ranking of algorithm $a_j \in \mathcal{A}$ at the testing experience t_i; ranking is here established according to the $\text{BPCER}_{0.001}$ for MAD. At each t_i, Borda count [26] is applied to score the tested algorithms, *i.e.* a decreasing number of points $p(r(a_j, t_i))$ is assigned to each algorithm based on the corresponding ranking, with $p(i) = |\mathcal{A}| - i$.

For each algorithm a_j, the points are accumulated over the different learning experiences and the total score is finally normalized by the maximum theoretical score:

$$\text{BRoT}(a_j) = \frac{\sum_{i=1}^{N} p(\mathcal{R}(a_j, t_i))}{|\mathcal{A}| \times N} \tag{6}$$

where N is the total number of testing experiences.

We observe the AUC metric is useful to understand the whole performance of the method across all the training experiences, highlighting the performance in terms of the best accuracy achieved. Differently, the BRoT metric enables the understanding of which algorithm has the greatest probability of having high accuracy across the whole learning process (without taking into consideration the possible gap between different methods in terms of absolute accuracy). These two metrics are complementary and help to understand different aspects of the performance of the investigated model in our novel scenario.

[1] https://github.com/serengil/deepface

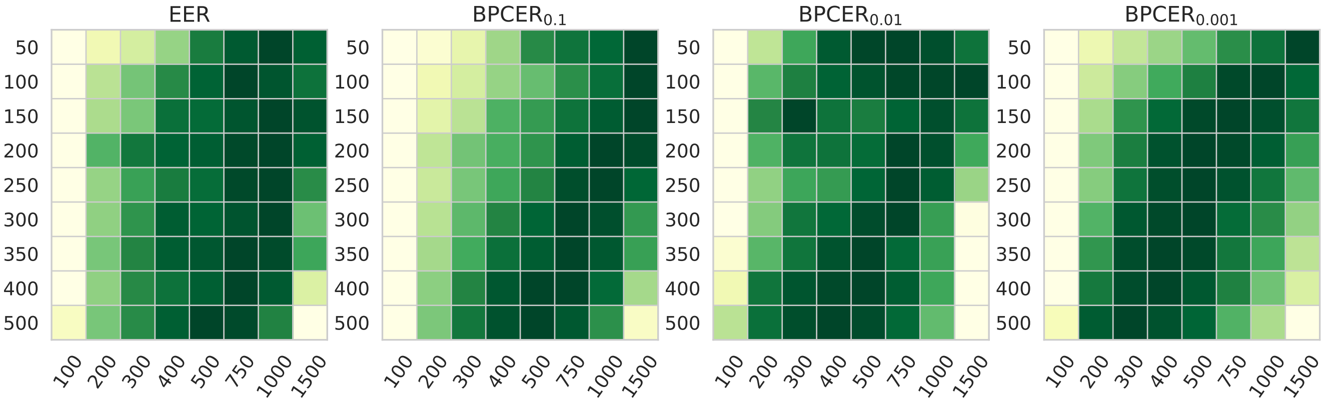


Figure 3. Performance of LwF [25] on the MAD task with respect to different training experience sizes (y-axis) and λ values (x-axis). Each matrix is referred to a specific metric commonly used in the MAD scenario, *i.e.* EER, BPCER$_{0.1}$, BPCER$_{0.01}$ and BPCER$_{0.001}$ (the most challenging case), as detailed in Section 4.3. The darker color is better. As shown, different λ values strongly impact the performance of the LwF algorithm, and suggest that larger values are the best choice for small experience sizes.

5. Results on MAD

5.1. Baselines

Different Continual Learning methods have been investigated. Elastic Weight Consolidation (**EWC**) [21] is based on a penalty loss that tries to constrain the model weights in maintaining the same value in new experiences. Learning without Forgetting (**LwF**) [25] contrasts the forgetting problem by exploiting two different models: the old model M_{t-1}, which is the result of training up to the current experience and thus carries the knowledge of previous data, and the current model M_t, which is initialized as a copy of the old model. The old model is frozen and is used through distillation to train the current model by adding a distillation component to the loss function. In this algorithm, an important hyperparameter is represented by λ, which acts as a regularization term used to balance the two components of the loss function. The value of the λ hyperparameter determines the trade-off between preserving knowledge from previous experiences and adapting to the new one: higher values of λ are used to emphasize the preservation of knowledge from previous experiences, while lower values favor the learning from current data. With $\lambda = 0$, no distillation happens and LwF collapses to a plain fine-tuning strategy. In LwF λ is fixed from the beginning and does not change between experiences. This choice may be suboptimal and an adaptive choice of λ may be beneficial, as demonstrated in the next sections.

We include in our analysis also the Synaptic Intelligence (**SI**) [51] method, based on a quadratic regularization that aims to preserve the weights that contribute to the performance on old data. Finally, we investigate the Deep Streaming Linear Discriminant Analysis (**SLDA**) [18] method that, taking inspiration from the data mining research field, uses a covariance matrix to perform the final prediction on pre-extracted features. This method, expressively developed for the Online Learning task [16], does not have a proper learning phase (intended as the learning process of common neural networks) and the concept of batch size, since it processes one sample per time in a streaming manner, without any memory mechanism.

	MAD ↓	
Exp. Size	**Small**	**Large**
Naive	+14%	+16%
EWC [21]	+14%	+16%
SI [51]	+21%	+24%
SLDA [18]	+27%	
LwF [25]	**+13%**	**+12%**

Table 1. Comparison of different CL methods with variable experience sizes. Results are expressed as the percentage variations of the AuC with respect to the ideal case, *i.e.* the Joint approach. A positive value indicates a higher error, lower values are desired.

In order to have a reference in results, we also implement two additional approaches: the first is the **naive** method, in which the model is optimized on available data, without any specific mechanism to contrast the catastrophic forgetting, while the second is the **Joint** method, in which all the training data are available at the beginning of the training procedure (*i.e.* the common Machine Learning scenario). All the compared baselines have been tested by exploiting the public implementations available in Avalanche [28].

5.2. Results

Firstly, we test baselines in our distributed training scenario with variable experience sizes. Results are reported in Table 1, for two different settings referred to as “**small**” and “**large**”, respectively; in the first one, the size of experience

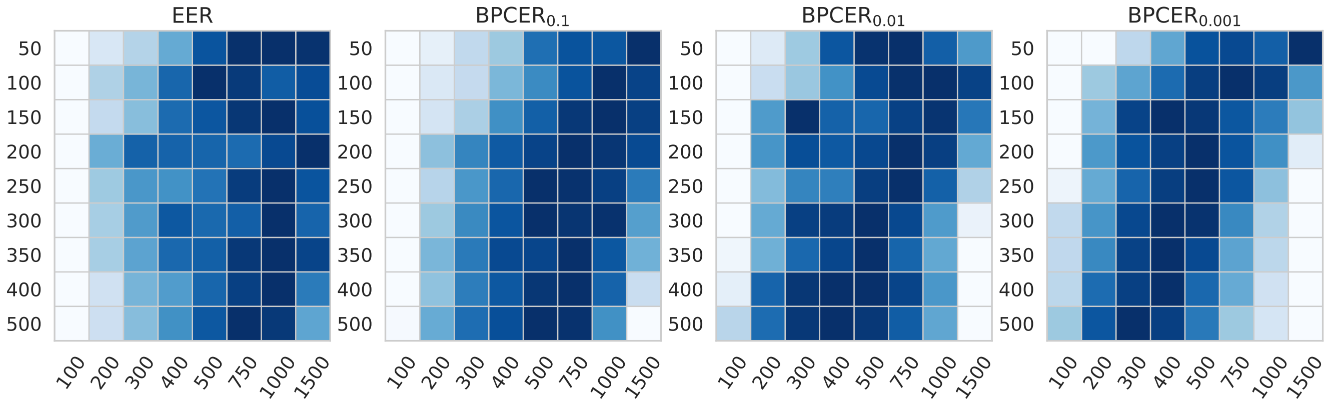


Figure 4. Performance of LwF [25] on the MAD task in terms of the BRoT metric (see Sect. 4.3), which highlights the probability to have the best performance across the whole distributed training process. The darker color is better.

varies in the range $[50, 500]$, and the probability to sample a specific size is modelled through the *Zipf*'s law [24]. In the second, the range is inverted $[500, 50]$, and the same distribution is exploited. In other words, in the case "small" there is a high probability to have small data chunks (of variable size) as input, and vice versa in the "large" one.

In this test, the upper bound is represented by the performance of the Joint training, in which the training set is available as a single chunk in a single site, and then results are expressed as the deviation percentage of the AUC metric (see Sect. 4.3) with respect to the one obtained with the Joint approach. Since AUC is related to errors and lower deviation values are better, results reveal that the distributed training scenario is challenging and that the size of small and large only partially impact the general performance. As expected, the SLDA method reports the same accuracy since it is not influenced by the experience size (see Sect. 5.1).

From a general point of view, the LwF approach tends to achieve the best performance and is therefore selected for the next investigation that is focused on the analysis of the weight assigned to the distillation loss with respect to the experience size. It is worth noting that this aspect is not yet fully investigated in the literature, especially in relation to different experience sizes.

In particular, we create data chunks with different fixed sizes, starting from 50 with a step of 50 ($|d| = \{50, 100, 150, ..., 500\}$). Then, differently from the previous case, the size of chunks is the same across the whole training procedure. For each defined size, we test different λ values: empirically, we found a specific range must be used to achieve reasonable results, in particular a range of $\{100, 1500\}$ for the MAD task. Results, obtained by averaging the test metrics on all possible dataset order configurations, are condensed through colored matrices (the darker color is better), as reported in Figures 3. We observe that different λ strongly impact the performance of the LwF, especially in relation to different training experience sizes. For instance, with $\lambda = 100$, the model performance is generally limited for all possible chunk sizes. From a general point of view, we observe a trend in which with small experience sizes it is better to have larger λ values, and vice versa. This tendency is particularly noticeable with BPCER$_{0.001}$ (last matrix of Fig. 3), which represents the most challenging (and realistic) case of the MAD task.

Then, we test LwF in terms of BRoT metric, reporting the results (expressed in the same visual form of colored matrices) in Figure 4. As shown, also the proposed BroT metric confirms the tendency noted in the previous cases, revealing that the proper choice of λ is needed to enhance the probability to achieve the best performance on the whole distributed training procedure.

Finally, we test MAD capabilities of the investigated solutions. Results are reported in Table 2, in which we show the performance of the following methods, all working on features extracted through the ArcFace architecture [11] from the two input images, and combined by a subtraction: i) the current state-of-the-art method described in [46], consisting in an SVM classifier with the Radial Basis Function kernel. The training scenario is Joint, since all data must be available before the single batch-based training procedure; ii) a solution equal to the previous one, but exploiting MLP as a classifier. This classifier has been adopted to simplify the comparison with the incremental training methods, in which we are forced to use a deep learning-based architecture to apply the CL strategies; iii) three incremental training approaches based on the investigated LwF method, trained with different experience sizes: large, small and the scenario with fixed experience that achieved the best performance (experience size equal to 500 and $\lambda = 200$).

As expected, the first method based on SVM and batch training exhibits the best performance in terms of EER and BPCER; such results represent in our experiments a sort of upper bound to the performance achievable in this scenario. Batch training, in fact, typically outperforms incremental

Training	Method	Classifier	Exp. Size	EER	BPCER$_{0.1}$	BPCER$_{0.01}$	BPCER$_{0.001}$
Batch	**ArcFace** [46]	SVM	Joint	0.121	0.135	0.275	0.528
	ArcFace [46]	MLP	Joint	0.138	0.209	0.722	0.825
Incremental	**LwF** [25]	MLP	small	0.156	0.294	0.837	0.837
	LwF [25]	MLP	large	0.145	0.230	0.786	0.922
	LwF [25]	MLP	fixed	0.140	0.224	0.679	0.896

Table 2. Experimental results obtained on the MorphDB dataset for the MAD task. In particular, it is possible to compare the performance of the sota MAD method [46], based on the common batch-based training (first row), with respect to the investigated LwF method for incremental training. In the Joint scenario, all training data are available at the same time, while "small", "large" and "fixed" refers to the incremental training described in Section 5.2.

learning and SVM proved to be the best classifier coupled with ArcFace features for the DMAD task. The MLP classifier unfortunately performs slightly worse, especially at BPCER levels corresponding to a low error threshold. Interestingly, all the investigated CL strategies have similar performance with respect to the batch-based training: then, we observe that LwF method is a promising method to bridge the gap between the common Machine Learning training scenario and the incremental training one, needed to deal with highly constrained scenarios.

	Classification ↑	
Exp. Size	**Small**	**Large**
Naive	-6.4%	-4.8%
EWC [21]	-5.7%	-5.1%
SI [51]	-9.2%	-5.2%
SLDA [18]	-2.7%	
LwF [25]	**-2.2%**	**-1.5%**

Table 3. Comparison of different Continual Learning methods with variable experience sizes. Results are expressed in terms of the percentage variations of the AuC with respect to the ideal case, *i.e.* the Joint approach. Note that a negative value indicates a lower accuracy, and then higher values are desired.

6. Further Investigation

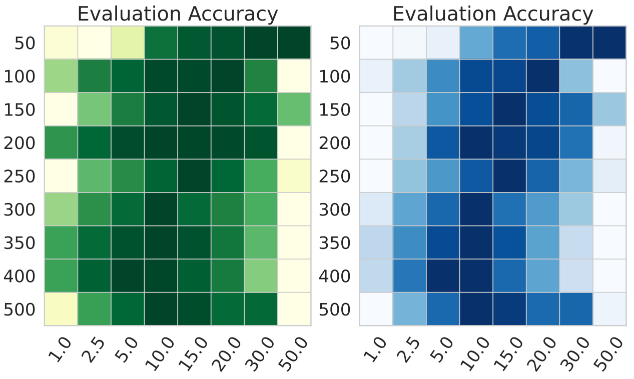
To validate our findings, we extend our investigation also on the supervised continual learning object classification task [47, 35], which is one of the most common tasks in the Continual Learning field. This task consists in continually training a classifier able to incrementally learn new instances, new classes, or both. In particular, to maintain similarity with the MAD task, we assume to work in the New Instances (NI) scenario, also referred to as Data Incremental [9], in which new instances of the same pre-defined classes become progressively available during the training phase. We observe this scenario is slightly different from the Domain Incremental (Domain-IL) [49] task, in which new instances belong also to different domains.

In our validation, we use **CORe50** [27] dataset, that contains 50 objects, belonging to 10 categories, acquired in 11 sessions (8 indoor and 3 outdoor) with different backgrounds. The dataset is organized as reported in the original paper, in which 8 sessions are used for training and validation and the remaining 3 for the testing procedure. The total amount of frame is about 164k with a resolution of 128×128 pixels. From the classification task point of view, this dataset is challenging due to changes in backgrounds (outdoor and indoor) and light sources, occlusions and low data variability during the same experience.

In order to reproduce the MAD training setting, we use a ResNet-50 [19] model, trained on the ImageNet [10] dataset, to extract features that are then classified by the same MLP architecture used in the MAD task. In this case, $|D| = 1$ and then we shuffle the dataset averaging the collected results on 10 runs. As metrics, we exploit the Top-1 accuracy, in which high values are positive. AUC and BroT metrics are based on obtained accuracy values across the testing experiences.

Results are reported in Table 3, in which large positive variations are better. We observe that the performance of the investigated CL methods is generally closer to the Joint baseline with respect to MAD task. LwF is confirmed as one of the best methods to limit the drop introduced by our distributed scenario, even though SLDA shows better behaviour in comparison to the MAD task.

Focusing our analysis on LwF, we create several chunks of data with a fixed size in order to analyze the impact of varying λ. Differently from MAD, empirical experiments suggest the proper range to achieve reasonable results is $\lambda = \{1, 50\}$. The visualization of the experimental results is reported in Figure 5a and 5b for the AUC and BRoT metrics, respectively. Results confirm our previous considerations on MAD task, *i.e.* lambda greatly impacts the performance of LwF, in particular with different sizes, even though this tendency seems to be less evident. In particular, $\lambda = 50$ is a proper choice only for the case in which the chunk size is equal to 50, while this value leads to a significant drop in performance in all the other cases.



(a) AUC

(b) BRoT

Figure 5. Performance of LwF [25] on the object classification task with respect to different training experience sizes (y-axis) and λ values (x-axis). The darker color is better.

7. Concluding Remarks

In this paper, we have carried out one of the first investigations about the performance of Continual Learning methods in an incremental and distributed training scenario addressing the Differential Morphing Attack Detection (D-MAD) task. In this scenario, in which data cannot be transferred between different sites due to privacy issues, the recent CL paradigm proves to be useful in enabling model transfer instead of data transfer.

Since data chunks available at each training experience may have different sizes in realistic usage, a further investigation has been conducted to analyze the proper parametrization for the LwF approach, an element not yet fully investigated in the CL literature. It is worth noting that, from a general point of view, the choice of the distillation loss value (λ) in the LwF approach is challenging, since it varies in relation to both experience size and task, and a wrong choice can lead to a significant drop in accuracy, as shown in the experimental evaluation.

The outcomes of our analysis can be summarized as follows: i) experimental results confirm the opportunity to use the Continual Learning paradigm, and specifically the LwF method, to train a MAD detector in a distributed and incremental manner in order to overcome privacy issues; ii) it clearly emerged, in view of future work, the need to automatically determine the proper parametrization of LwF, in terms of the value of λ with respect to the size of the training chunk, following the general consideration that small experience size should need larger values; iii) it is also important to note that further analysis is needed in order to properly determine the λ ranges in relation to a specific dataset or task to be addressed; iv) additional research investigations are important to improve the final accuracy of the MAD model trained in the incremental and distributed setting, that still suffers in terms of performance with respect to a model trained in the common Machine Learning setting (batch training on the whole dataset);.

In conclusion, we believe that our findings can be useful in future research work in the field of Morphing Attack Detection, in order to enable distributed and incremental training, overcome privacy issues and train new models on more varied and large datasets, but also the Continual Learning task, to properly define values of the λ parameter taking into consideration the size of the training experience.

Acknowledgment

This work is part of the iMARS project. The project received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 883356. Disclaimer: this text reflects only the author's views, and the Commission is not liable for any use that may be made of the information contained therein.

References

[1] H. Ahn, S. Cha, D. Lee, and T. Moon. Uncertainty-based continual learning with adaptive regularization. *Advances in neural information processing systems*, 32, 2019. 3

[2] E. Bisong and E. Bisong. Batch vs. online learning. *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, pages 199–201, 2019. 1

[3] G. Borghi, G. Graffieti, A. Franco, and D. Maltoni. Incremental training of face morphing detectors. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 914–921. IEEE, 2022. 1, 3

[4] G. Borghi, E. Pancisi, M. Ferrara, and D. Maltoni. A double siamese framework for differential morphing attack detection. *Sensors*, 21(10):3466, 2021. 3

[5] G. Borghi, S. Pini, F. Grazioli, R. Vezzani, R. Cucchiara, et al. Face verification from depth using privileged information. In *BMVC*, page 303, 2018. 2

[6] P. Buzzega, M. Boschini, A. Porrello, and S. Calderara. Rethinking experience replay: a bag of tricks for continual learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2180–2187. IEEE, 2021. 2

[7] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE, 2018. 4

[8] L. DeBruine and B. Jones. Face research lab london set. *Psychol. Methodol. Des. Anal*, 2017. 3

[9] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 7

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database.

In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7

[11] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 4, 6

[12] B. Dorizzi, R. Cappelli, M. Ferrara, D. Maio, D. Maltoni, N. Houmani, S. Garcia-Salicetti, and A. Mayoue. Fingerprint and on-line signature verification competitions at icb 2009. In *Advances in Biometrics: Third International Conference, ICB 2009, Alghero, Italy, June 2-5, 2009. Proceedings 3*, pages 725–732. Springer, 2009. 4

[13] A. Douillard, A. Ramé, G. Couairon, and M. Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9285–9295, 2022. 3

[14] M. Ferrara, A. Franco, and D. Maltoni. The magic passport. In *IEEE International Joint Conference on Biometrics, Clearwater, IJCB 2014, FL, USA, September 29 - October 2, 2014*, pages 1–7. IEEE, 2014. 1, 2

[15] M. Ferrara, A. Franco, and D. Maltoni. Face demorphing. *IEEE Transactions on Information Forensics and Security*, 13(4):1008–1017, 2017. 4

[16] G. Graffieti, G. Borghi, and D. Maltoni. Continual learning in real-life applications. *IEEE Robotics and Automation Letters*, 7(3):6195–6202, 2022. 3, 4, 5

[17] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 87–102. Springer, 2016. 4

[18] T. L. Hayes, N. D. Cahill, and C. Kanan. Memory efficient experience replay for streaming learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9769–9776. IEEE, 2019. 5, 7

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[20] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 3

[21] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13), 2017. 5, 7

[22] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016. 1

[23] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020. 1

[24] W. Li. Zipf's law everywhere. *Glottometrics*, 5(2002):14–21, 2002. 6

[25] Z. Li and D. Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 2, 5, 6, 7, 8

[26] D. Lippman. Voting theory. *Creative Commons BYSA*, 2013. 4

[27] V. Lomonaco and D. Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 17–26, 2017. 7

[28] V. Lomonaco, L. Pellegrini, A. Cossu, A. Carta, G. Graffieti, T. L. Hayes, M. D. Lange, M. Masana, J. Pomponi, G. van de Ven, M. Mundt, Q. She, K. Cooper, J. Forest, E. Belouadah, S. Calderara, G. I. Parisi, F. Cuzzolin, A. Tolias, S. Scardapane, L. Antiga, S. Amhad, A. Popescu, C. Kanan, J. van de Weijer, T. Tuytelaars, D. Bacciu, and D. Maltoni. Avalanche: an end-to-end library for continual learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2nd Continual Learning in Computer Vision Workshop, 2021. 5

[29] S. Madhavan and N. Kumar. Incremental methods in face recognition: a survey. *Artificial Intelligence Review*, 54(1):253–303, 2021. 1

[30] A. Martinez and R. Benavente. The ar face database: Cvc technical report, 24, 1998. 4

[31] M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 2, 3

[32] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 2

[33] T. Neubert, A. Makrushin, M. Hildebrandt, C. Kraetzer, and J. Dittmann. Extended stirtrace benchmarking of biometric and forensic qualities of morphed face images. *IET Biometrics*, 7(4):325–332, 2018. 3

[34] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. 1, 3

[35] G. Pasquale, C. Ciliberto, F. Odone, L. Rosasco, and L. Natale. Are we done with object recognition? the icub robot's perspective. *Robotics and Autonomous Systems*, 112:260–281, 2019. 7

[36] L. Pellegrini, G. Graffieti, V. Lomonaco, and D. Maltoni. Latent replay for real-time continual learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10203–10209. IEEE, 2020. 2

[37] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 947–954. IEEE, 2005. 3, 4

[38] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss. The FERET database and evaluation procedure for face-

recognition algorithms. *Image and vision computing*, 16(5):295–306, 1998. 3, 4

[39] A. Quek. FaceMorpher morphing algorithm. https://github.com/alyssaq/face_morpher. Accessed: 2022-11-30. 3

[40] K. Raja, M. Ferrara, A. Franco, L. Spreeuwers, I. Batskos, F. de Wit, M. Gomez-Barrero, U. Scherhag, D. Fischer, S. K. Venkatesh, et al. Morphing attack detection-database, evaluation platform, and benchmarking. *IEEE transactions on information forensics and security*, 16:4336–4351, 2020. 2, 4

[41] E. Sarkar, P. Korshunov, L. Colbois, and S. Marcel. Vulnerability analysis of face morphing attacks from landmarks and generative adversarial networks. *arXiv preprint arXiv:2012.05344*, 2020. 3

[42] E. Sarkar, P. Korshunov, L. Colbois, and S. Marcel. Are gan-based morphs threatening face recognition? In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2959–2963. IEEE, 2022. 3

[43] Satya Mallick. "Face morph using opencv — c++ / python. https://learnopencv.com/face-morph-using-opencv-cpp-python/. Accessed: 2022-11-30. 3

[44] U. Scherhag, A. Nautsch, C. Rathgeb, M. Gomez-Barrero, R. N. Veldhuis, L. Spreeuwers, M. Schils, D. Maltoni, P. Grother, S. Marcel, et al. Biometric systems under morphing attacks: Assessment of morphing techniques and vulnerability reporting. In *2017 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–7. IEEE, 2017. 2

[45] U. Scherhag, C. Rathgeb, J. Merkle, R. Breithaupt, and C. Busch. Face recognition systems under morphing attacks: A survey. *IEEE Access*, 7:23012–23026, 2019. 2

[46] U. Scherhag, C. Rathgeb, J. Merkle, and C. Busch. Deep face representations for differential morphing attack detection. *IEEE transactions on information forensics and security*, 15:3625–3639, 2020. 4, 6, 7

[47] Q. She, F. Feng, X. Hao, Q. Yang, C. Lan, V. Lomonaco, X. Shi, Z. Wang, Y. Guo, Y. Zhang, F. Qiao, and R. H. M. Chan. Openloris-object: A robotic vision dataset and benchmark for lifelong deep learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4767–4773, 2020. 7

[48] G. M. van de Ven, H. T. Siegelmann, and A. S. Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1):1–14, 2020. 3

[49] G. M. Van de Ven and A. S. Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 7

[50] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 4

[51] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017. 5, 7