Pattern Recognition Letters 180 (2024) 82–89

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Continual learning for adaptive social network identification

Simone Magistri *, Daniele Baracchi, Dasara Shullani, Andrew D. Bagdanov, Alessandro Piva

Department of Information Engineering, University of Florence, Via di S. Marta 3, 50134, Firenze, Italy

ARTICLE INFO

Editor: Maria De Marsico

Keywords:
Continual learning
Social network identification
Multimedia forensics

ABSTRACT

The popularity of social networks as primary mediums for sharing visual content has made it crucial for forensic experts to identify the original platform of multimedia content. Various methods address this challenge, but the constant emergence of new platforms and updates to existing ones often render forensic tools ineffective shortly after release. This necessitates the regular updating of methods and models, which can be particularly cumbersome for techniques based on neural networks which cannot quickly adapt to new classes without sacrificing performance on previously learned ones – a phenomenon known as *catastrophic forgetting*. Recently, researchers aimed at mitigating this problem via a family of techniques known as *continual learning*. In this paper we study the applicability of continual learning techniques to the social network identification task by evaluating two relevant forensic scenarios: *Incremental Social Platform Classification*, for handling newly introduced social media platforms, and *Incremental Social Version Classification*, for addressing updated versions of a set of existing social networks. We perform an extensive experimental evaluation of a variety of continual learning approaches applied to these two scenarios. Experimental results demonstrate that, although Continual Social Network Identification remains a difficult problem, catastrophic forgetting can be significantly mitigated in both scenarios by retaining only a fraction of the image patches from past task training samples or by employing previous tasks prototypes.

1. Introduction

Multimedia content such as images and videos have become one of the primary means by which information is shared between Internet users. Unfortunately, this also includes content used to perpetrate crimes such as cyber bullying, incitement to hatred, and revenge porn. As a result, determining the origin of multimedia content is of great interest not only to law enforcement agencies but also to the general public. As the number of images and videos stored in seized devices can easily reach into the thousands, such analysis can often only be performed by automatic tools. This problem has been addressed by the multimedia forensics community through a number of techniques capable of analyzing different aspects of content history. Among these, discovering the social network from which content was downloaded has become of great interest in the last few years [1]. Knowledge about the social network of origin can then be used to guide further analyses, which can ultimately lead to the complete reconstruction of content history.

Unfortunately, the identification of social networks is a daunting task due to their black box nature. Their inner workings are closely guarded by parent companies who consider them proprietary information and researchers are consequently forced to depend on hidden clues embedded in shared media that arise from the processing performed by social platforms. The processing chain that multimedia content undergoes ends with a compression algorithm to reduce file size as much as possible while maintaining maximum visual quality [2]. When a picture is taken, the vast majority of smartphones and cameras store the resulting file in JPEG format. A similar procedure, which can also include resizing, renaming, and editing all or part of the metadata [3], occurs when content is shared on a social platform, resulting in a *double* JPEG compression trace. Numerous studies propose methodologies for Social Network Identification (SNI) that rely on factors such as JPEG quantization tables, pixel resolution, and image metadata [4]. Some researchers exploit the distribution of Discrete Cosine Transform (DCT) coefficients [5–8] as well as Discrete Wavelet Transform coefficients (DWT) [9]. Moreover, the distinctive fingerprint of Photo Response Non-Uniformity (PRNU) noise [10], renowned for its camera ballistics capabilities, has also been taken into account for image-based SNI [11–13]. Researchers have also investigated the importance of the container structure of multimedia content [14–17] to detect a specific SN platform in the content history. Today, the task of social network identification (single or multiple) is addressed predominantly through

* Corresponding author.
E-mail addresses: simone.magistri@unifi.it (S. Magistri), daniele.baracchi@unifi.it (D. Baracchi), dasara.shullani@unifi.it (D. Shullani), andrew.bagdanov@unifi.it (A.D. Bagdanov), alessandro.piva@unifi.it (A. Piva).

https://doi.org/10.1016/j.patrec.2024.02.020
Received 3 July 2023; Received in revised form 2 January 2024; Accepted 24 February 2024
Available online 28 February 2024

deep learning techniques based on Convolutional Neural Networks (CNNs) [1,18], using all or a combination of the above-mentioned fingerprints to exploit both spatial and meta information of the content itself.

Despite the remarkable SNI results reported for existing CNN-based methods, the task of keeping them current and effective poses a significant challenge due to the ever-changing nature of the social network landscape. Indeed, as companies fiercely compete to attract new users to their platforms, the software responsible for managing these networks undergoes constant updates to incorporate new features and enhance existing ones. This, in turn, leads to modifications in the traces left on shared content, consequently requiring the update of previously trained models. Additionally, new economic players consistently strive to enter the market by proposing new platforms, hoping to address gaps in existing products and establish themselves in this growing industry. As data-driven methods are usually designed to classify among a predetermined set of possibilities, incorporating additional platforms inevitably requires training a new model. Because of the phenomenon known as *catastrophic forgetting*, existing models cannot be easily updated by solely finetuning them on new data; indeed, when a CNN is initially trained on one task and subsequently trained on one or more new tasks, it quickly loses its ability to perform the initial task [19]. A naive solution to avoiding catastrophic forgetting, called Joint Incremental Training, consists of jointly training the network on the new data along with the old ones. The main problem with joint training is that it is expensive to re-train the network with the entire dataset each time new data become available. Furthermore, it may not always be possible to retrieve data from previous tasks due to privacy considerations or because they are simply no longer available.

Continual Learning (or Incremental Learning) approaches strive to reduce catastrophic forgetting by making efficient use of limited data from past tasks. In the context of multimedia forensics, a first attempt at applying continual learning techniques (although not for social network identification) was performed by Marra et al. [20]. This work showcased the efficacy of iCaRL [21] in expanding the capabilities of a network for GAN-generated image identification. Early work on applying continual learning for SNI was done by Magistri et al. [22], who introduced an effective convolutional architecture that was shown to be extensible to new social platforms.

In this paper we extend the findings of Magistri et al. [22], whose study focused on updating a model to accurately classify *newly introduced* social media platforms. We denote this scenario as *Incremental Social Platform Classification* (ISPC). Here we introduce a more challenging task in which we must update a model in order to accommodate *versions* of the original set of social networks. This update entails not only the ability to handle these new versions but also to differentiate between them. We denote this scenario as *Incremental Social Version Classification* (ISVC). We perform an extensive experimental evaluation of the techniques used by Magistri et al. [22], as well as three new state-of-the-art methods, on both scenarios. Additionally, we investigate how the number of exemplars from past tasks affects the results of exemplar-based techniques. Our experiments demonstrate that, by employing a limited memory budget of image patches, existing continual learning methods can approach *Joint Incremental Training* performance in both ISPC and ISVC scenarios.

2. Continual learning

In this section we introduce the formulation of the Continual Learning problem and discuss works from the literature most related to our contributions.

2.1. Continual learning scenarios

Typically, a Convolutional Neural Network model $\mathcal{M}$ designed for classification consists of two key components: a feature extractor parameterized by θ, which processes input $x \in \mathcal{X}$ and produces a representation $z = f(x;\theta)$, and a classification head $g(z;W)$, parameterized by W responsible for classifying the input into a set of predefined categories.

In a continual learning scenario, the model $\mathcal{M}$ undergoes sequential training on a collection of T disjoint classification tasks, denoted as $\mathcal{D} = \{(\mathcal{X}_t, C_t)\}_{t=1}^{T}$, where $C_t \cap C_{t'} = \emptyset$ for $t \neq t'$. Each task t consists of a set of input samples $\mathcal{X}_t$ and their associated labels C_t. For each incremental learning task t, the model is trained to accurately classify the class C_t. This is accomplished by introducing a classification head W_t dedicated to task t. The optimizer jointly trains the heads $\{W'_t\}_{t'=1}^{t}$ and feature extractor weights θ during this step. At the end of task T, the network should be capable of classifying classes from all seen tasks $t = 1, \dots, T$.

Continual learning seeks to mitigate catastrophic forgetting by introducing a regularizer into the network training objective in order to preserve performance on previous tasks. A general structure of a training loss for continual learning is:

$$L_t = L_t^{CE} + \lambda_{reg} L_t^{reg} \tag{1}$$

where L_t^{CE} is the cross entropy loss for task t, L_t^{reg} is a regularization loss aimed at reducing catastrophic forgetting, and $\lambda_{\text{reg}} \in \mathbb{R}$ is a hyperparameter balancing the two losses. In the next section we provide an overview of continual learning methods and describe some common regularization losses.

2.2. Related work

Continual learning methods can be roughly grouped into two macro-categories: *exemplar-free* approaches [23–25] which do not store exemplars from past tasks and only add extra terms to the training loss to incorporate knowledge from past tasks during the training of new ones, and *exemplar-based* approaches [21,26,27], which rely on a small subset of representative samples (exemplars) from previous tasks.

The main goal of exemplar-free methods is to reduce catastrophic forgetting with the assumption that samples from previous task cannot be stored due to privacy regulations or data security constraints. Examples of exemplar-free methods include Elastic Weight Consolidation (EWC) [23] and Riemannian Walk (RWalk) [24] which define a weight regularization term L_t^{reg} based on the Fisher Information Matrix to prevent network weights from drifting away from the previous task model when learning new task classes. Learning without Forgetting (LwF), instead, uses Knowledge Distillation [28] to discourage predictions from drifting when learning new tasks [25].

Knowledge Distillation (KD) has been employed as a regularization technique by many exemplar-based methods [21,26,27,29]. Moreover, exemplar-based approaches place a significant emphasis on tackling the challenge of imbalanced data between exemplars and current-task data. The imbalance between the number of exemplars from past task classes and number of training samples for current-task classes results in a task-recency bias towards classifying images into classes of the current task [30]. To mitigate this task-recency bias, methods such as Bias Correction (BIC) [26] and Incremental Learning With Dual Memory (IL2M) [27] rectify the network outputs. More recently, the SS-IL [29] was proposed which employs a separated softmax output layer in combination with task-wise knowledge distillation in order to reduce task-recency bias. Techniques like Incremental Classifier and Representation Learning (iCaRL) [21] avoid this bias by using a nearest-mean rule in feature space for classification instead of relying on classification heads trained with the cross-entropy loss.

The primary challenge faced by initial attempts at exemplar-free methods lies in their inability to mitigate task-recency bias due to

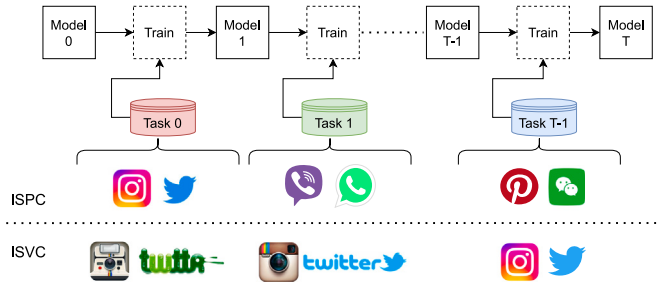

Fig. 1. The Incremental Social Platform Classification (ISPC) and Incremental Social Version Classification (ISVC) scenarios. See Section 3 for details.

absence of exemplars. Recent advancements have introduced *prototype rehearsal* as a method to leverage past-task deep features, enhancing the final classifier output and thereby mitigating task-recency bias. Prototypes, or class-means, are computed as the averages of feature vectors for each class in previous tasks. The storage of these prototypes aligns with the privacy requirements inherent in exemplar-free methods. In FeTrIL [31] a fixed feature extractor was proposed and the training is performed only on the last classifier using both previous task prototypes and current task features. EFC [32] suggests using a Prototype Rehearsal Asymmetric Cross-entropy loss (PR-ACE) along with the Empirical Feature Matrix (EFM) to selectively regularize feature space drift and prevent catastrophic forgetting while maintaining enough plasticity to still learn new tasks.

3. Continual SNI scenarios and model architecture

Given the perpetual state of change and evolution in the social network landscape, we believe that the application of continual learning techniques can significantly enhance social network identification systems. A notable advantage of Continual SNI methods is that they eliminate the need for maintaining a continually expanding dataset containing both old and new data. Such approaches would address concerns related to efficiency and privacy, as managing massive datasets is complicated and sensitive content need not be retained indefinitely. Additionally, the capability to update a model by training it solely on new data would offer significant time-efficiency advantages compared to retraining the entire model from scratch, thus making the process of building an updated model more cost-effective and energy-efficient.

3.1. Two scenarios for continual social network identification

To demonstrate these advantages, we envision two practical scenarios arising from real-world social network identification tasks. In the first scenario, which we call *Incremental Social Platform Classification* (ISPC), we hypothesize the emergence of new social networks over time. Since existing models could not have possibly been trained on these new platforms, they are bound to misclassify content coming from them, associating images with one of the pre-existing social networks. In this case our goal is to update the model to make it capable of classifying both the platforms on which it was originally trained on as well as newly-introduced ones.

In the second scenario, which we call *Incremental Social Version Classification* (ISVC), we hypothesize the release of new versions of existing social networks. These updates may significantly alter the processing pipeline used to produce media content, leading to a drop in classification accuracy for models trained on older datasets. We therefore have the aim to update the network to make it correctly classify both media content produced by older version of the available social platforms and media content produced by the updated versions. Moreover, by modifying the model to make it capable of classifying both the social platform and its version, we could leverage this additional information as a clue on the temporal origin of the content.

We give a pictorial representation of the two proposed scenarios in Fig. 1. In the ISVC scenario the goal is to classify images according to one of two possible social platforms (e.g. Instagram, Twitter) from which they were downloaded, and we assume that those platforms undergo updates over time to incorporate new features. In this case, the first task involves classifying images from the original versions of the social networks. The second task entails classifying images from the first update of those platforms, and so on for subsequent tasks. For ISPC, on the other hand, we assume that completely new social platforms are introduced over time. In this case, the primary task involves classifying images from an initial set of social platforms (e.g. Instagram, Twitter). The second task then entails classifying images from the new platforms (e.g. Viber, WhatsApp), and each subsequent task then involves handling an additional set of social networks. In both scenarios, our goal is to update a classifier to handle subsequent tasks while still retaining its ability to classify the previous ones.

3.2. Model architecture

We use the SNI architecture first presented in the study conducted by Magistri et al. [22], which is depicted for completeness in Fig. 2. The dual-branch network is inspired by the one proposed by Amerini et al. [13]. Due to the fixed-sized input requirement of the neural network, images are divided into non-overlapping patches of 256×256 pixels. This partitioning allows for inclusion of images with different resolutions, eliminating the need for resizing operations which can introduce artifacts caused by the subsampling algorithm and inadvertently erase the subtle cues left by the social network.

Instead of using image patches directly, we first perform a preprocessing step to produce as input to the network two complementary representations of the image signal. In the first representation, each patch x is split in non-overlapping 8×8 pixel blocks aligned with the JPEG grid. The first 9 quantized AC DCT components of each block are then used to build 9 histograms representing values between -50 and 50, which are then concatenated in a feature vector h. For the second representation, the original patch x is transformed into a residual image $\hat{x}$ by means of a high-pass filter which discards DCT coefficients corresponding to the lowest 1250 frequencies. The rationale for this is that recompression traces are usually left in medium frequencies, so the two strategies respectively enhance low/medium or medium/high frequencies.

The DCT histogram representation is processed by the *Histogram Branch*, a classical feed-forward neural network composed of three ReLU layers (of 512, 256, and 256 neurons) interleaved with two dropout layers. The high-pass patch representation is processed by the *Convolutional Branch*, a ResNet-18 [33] backbone which has been shown to be a good convolutional backbone for multiple tasks. Representations learned by the two branches are then concatenated and fused using a single ReLU layer with 512 neurons, producing a feature vector z. Finally, z is fed into several classification heads, each corresponding to a different task in continual SNI scenarios, to generate output probabilities.

4. Experimental setup and training procedure

In this section we describe the experimental protocol, our proposed dataset splits, and the implementation we use for our experiments on continual SNI.

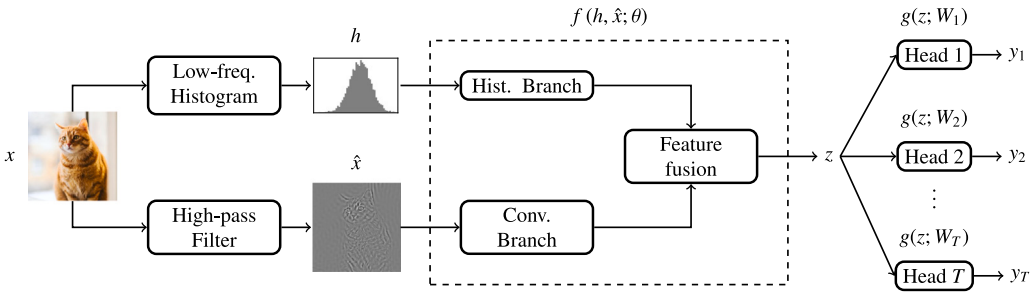

Fig. 2. Model architecture for Continual SNI [22]. The network consists of two parallel branches: the top branch is a multilayer perceptron taking as input the histograms of quantized DCT coefficients, and the lower branch is a ReNet-18 backbone taking as input an image after high-pass filtering. The two representations are fused via concatenation before the classification heads. See Section 3.2 for details.

Table 1
Smartphone Images and Social Update dataset statistics.

Dataset	#Classes	Split	#Patches	#Images	#Devices
Smartphone Images	14	Train	462k	17k	13
		Valid	92k	2k	2
		Test	131k	5k	4
Social Update	4/15	Train	574k	26k	52
		Valid	78k	3k	7
		Test	138k	6k	13

4.1. Datasets

To assess the effectiveness of continual learning approaches in ISPC and ISVC scenarios, we require datasets characterized by a substantial number of classes for task partitioning. In the ISPC scenario, we use the **Smartphone Images (SI) dataset** [34]. This dataset consists of a variety of indoor and outdoor images captured by multiple smartphones. Similarly to Magistri et al. [22], we consider 14 social networks split into 4 tasks. Specifically, we allocated five social networks for the first task and three for each of the remaining three tasks. We do not use a fixed class order for the tasks, but we use different random seeds for each run in order to reduce the bias induced by the choice of class ordering [30].

For the ISVC scenario, we collected a dataset, called **Social Update (SU)**, which contains different versions of four major social platforms: Facebook, Instagram, Twitter, WhatsApp. The SU dataset was created by gathering images from multiple datasets including SI, IPLAB [4] and FODB [35]. These datasets consist of both indoor and outdoor scenes, captured using smartphones and cameras, and shared on social media. Moreover, we incorporate SocialNews [36], a dataset consisting of images shared by news organizations and influencers on social networks. From this last dataset we only have images from Facebook, Instagram, and Twitter since WhatsApp was not available. As a result, the dataset it characterized by 4 social network platforms and a total of 15 versions. We adopted a specific task division where each task focuses on a particular social network version, corresponding to a specific dataset. The tasks are ordered based on the chronological sequence of their publication date. The first task is based on the SI dataset (containing data released in 2015), followed by the IPLAB dataset (2016), the FODB dataset (2021), and finally the SocialNews dataset (2023). As a result, the first three tasks consist of four social networks each, while the last task only includes three. We use the SU dataset for two distinct objectives. The first is to assess the capability of an incrementally trained network to accurately classify a social network after an update, which entails a 4-class classification problem. The second objective entails evaluating the performance of the network in terms of social version classification, which is a 15-class classification problem.

To ensure a fair evaluation and eliminate any biases stemming from the acquisition device, we divided the SI and SU dataset into three separate sets (training, validation, and test). These sets were carefully constructed to ensure there is no overlap in the devices used. In Table 1 we report the overall statistics of the two datasets.

4.2. Task agnostic performance metric

In the context of continual learning, where multiple tasks $t = 1, \dots, T$ are sequentially trained, we require a metric quantifying the performance deterioration caused by catastrophic forgetting across these tasks. The average accuracy A_T on the seen tasks up to the last task T is a common metric to assess the overall performance:

$$A_T = \frac{1}{T} \sum_{i=1}^{T} a_{i,T} \tag{2}$$

where $a_{i,T}$ is the accuracy obtained on task i after learning task T. We measure this metric for both patches and images in *Task Agnostic* (TAG) setting, assuming that task identifier is not known at inference time. As outlined in the preliminary version of this work [22], this scenario is more challenging and realistic for performance evaluation in continual SNI, if compared to *Task Aware* (TAW) setting where the task identifier is given.

4.3. Training and test settings

We ran all experiments using FACIL [30], a continual learning framework for PyTorch. We used the default hyperparameters of FACIL for all tested approaches, except for EWC where we use $\lambda_{reg} = 500$ to give less weight to the regularizer, and for LwF and BIC where we set the KD temperature [28] to $T = 1$. See the FACIL paper for more details on the hyperparameters of these approaches [30]. In addition we implemented three recent approaches (SS-IL [29], FeTrIL [31], EFC [32]) that were not available in FACIL. For these methods, we used the hyper-parameters provided by the authors. All experiments were run five times initializing the weights with different random seeds and, for the ISPC scenario, randomizing class order.

For each task, we trained our model from scratch (i.e. we did not use a pretrained ResNet-18 for the convolutional branch) using Adam [37] with an initial learning rate of 10^{-3} which was decayed when the validation loss on the current task did not improve for 20 epochs. Training was stopped when the learning rate reached 10^{-6} or when 200 epochs were reached. For each epoch, we randomly sampled one crop per image in order to reduce the training time. All patches were evaluated during the validation and test phases.[1]

Moreover, all the evaluated continual learning approaches except iCaRL compute the global decision by averaging the softmax outputs for patches. For iCaRL, after training each task, the feature vectors belonging to every patch class are extracted and their mean is computed. At inference time, test patches are classified according to the nearest class mean, while images are classified according to the minimum average distance of their patches to the class mean.

[1] Code is available at https://github.com/simomagi/continual_SNI.

Table 2
Comparison on Smartphone Images with the state-of-the-art. The proposed architecture obtains a higher accuracy both on patch- and image-level classification problems. At the same time, our network is smaller (with respect to the number of parameters) compared to the model proposed for SNI in [13].

Method	Patch accuracy ↑	Image Accuracy ↑		# Parameters ↓
		Avg softmax	Majority vote	
Amerini et al. [13]	63.1 (±1.2)	72.9 (±1.6)	71.7 (±1.4)	73.3 M
Proposed architecture	**64.6** (±1.2)	**75.4** (±0.6)	**74.7** (±0.8)	**12.2 M**

Table 3
Performance comparison on Smartphone Images using only the convolutional branch after preprocessing with different filters.

Image pre-processing	Accuracy ↑	
	Patch	Image
No filter	38.1 (±6.0)	48.3 (±7.5)
Mihcak Filter [38]	49.7 (±2.8)	62.6 (±3.7)
Ours (DCT High-pass filter)	**55.1** (±2.9)	**67.9** (±3.2)

5. Experimental results and discussion

In this section we discuss the effectiveness of the proposed architecture on both standard and two continual learning scenarios (ISPC and ISVC). For each continual scenario, the lower-bound (**LB**) baseline for comparison is *Finetuning* which simply consists of training the network on the new task data, while the upper bound (**UB**) is *Joint Incremental* training which consists of re-training the network on new task data along with all data from previous tasks.

For exemplar-based methods we used a fixed-size memory $\mathcal{M}$ with a capacity of K, containing randomly sampled image patches. We chose to save only patches and not entire high resolution images for two reasons. Firstly, since the patches have a fixed size of 256×256, they incur a lower memory burden. Secondly, saving only patches can be useful for applications where the full image content cannot be saved due to privacy concerns. After each new task, we use a rebalancing procedure for the patches stored in $\mathcal{M}$. We randomly discard patches from previous tasks to ensure a uniform distribution of exemplar patches per class. By ensuring this uniform distribution, we maintain a constant overall memory dimension K.

5.1. The effectiveness of the SNI architecture

To validate the effectiveness of our architecture, we trained it on all 14 Smartphone Image classes and compared results with the method of Amerini et al. [13]. We present image-level performance using both patch majority voting (as in [13]) and averaged patch softmax predictions. Our model outperforms theirs in both image and patch classification (see Table 2), demonstrating superior accuracy with reduced complexity.

We also evaluated different image preprocessing filters for the convolutional branch, comparing non-filtered images, those filtered using [38] (as in [13]), and our proposed high-pass filter. Results (see Table 3) highlight the performance improvement achieved through image filtering, with our DCT-based method yielding improved accuracy for the convolutional branch.

5.2. Incremental social platform classification (ISPC)

In Table 4 we report performance in average accuracy after the last task. We compare both exemplar-free and exemplar-based methods as originally reported in [22]. In addition, three new methods are evaluated, FeTrIL and EFC which are exemplar-free and SS-IL which uses exemplars. We also provide results for exemplar-based extensions of EWC, LwF, and RWalk in which a small number of training samples for each task are retained as exemplars and replayed when training on a new one. Most exemplar-free techniques fail to achieve satisfactory performance, demonstrating only modest improvement over Finetuning. Only two approaches (EFC and FeTrIL) are capable of significantly increasing accuracy with respect to the lower bound. This outcome comes from the incorporation of prototypes, which are reintroduced during training to strike a balance between the previous and current task classifiers, effectively alleviating task-recency bias. In the exemplar-based setting, by incorporating $K = 500$ exemplars, iCaRL and BIC are the top performing methods reducing the performance gap with Joint Incremental by half. Note that the relatively high standard deviations are due to the random ordering of classes, as certain social networks exhibit similar characteristics.

5.3. Incremental social version classification (ISVC)

In Tables 5 and 6 we report performance for both exemplar-free and exemplar-based methods on the ISVC scenario for 15 and 4 classes respectively. In the 15-class setting, each social network platform *version* is treated as a separate class. In the 4-class setting, on the other hand, examples are only labeled with the originating social platform without considering the specific version. We emphasize that we did not train separate models for the 4-class case. Instead, we obtained the results by performing *a posteriori* remapping of the network outputs and disregarding any information pertaining to the version. As expected, predicting the social network version along its type (ISVC-15 classes) is a significantly more challenging setting. Indeed, most methods show a drop of more than 20 points in TAG accuracy with respect to ISVC-4 classes.

Moreover, we highlight that in the ISVC scenario the two prototype-based exemplar-free methods (FeTrIL and EFC) manage to achieve superior results when compared to exemplar-based approaches (with memory size $K = 500$). This outcome underscores the efficacy of prototypes as a viable solution when it is not possible to store exemplars from previous tasks. Notably, EFC emerges as the top-performing method, reducing the performance gap in image classification with Joint Incremental by approximately 14%.

5.4. Closing the gap with joint-training

In this section we investigate the impact of memory size K on exemplar-based approaches and compare performance with FeTril and EFC, which are exemplar-free but achieved competitive results in the ISVC scenario. Moreover, we examine how far current continual learning solutions are from joint incremental training (**UB**). In Fig. 3, we give the performance of all approaches in all the SNI scenarios for K ranging from 100 to 2000.

Difference in results between ISPC 14-class and ISVC 15-class scenarios is to be expected. Indeed, while the number of classes is comparable in both scenarios, there is a key distinction. In the first case, each class represents an entirely different social network, whereas in the second case multiple classes represent different versions of the same platform. Moreover, while ISPC 14-classes only accounts for a single dataset with fairly homogeneous data, ISVC 15-classes uses a mix of four different datasets acquired using multiple devices and following different protocols. iCaRL and BIC consistently outperform the other approaches in all the considered scenarios. SS-IL obtains competitive performance in ISVC scenario, while it obtains poor results for the

Table 4
Incremental Social Platform Classification (ISPC) on 14 classes in Average TAG accuracy on patches and images after the last task with and without exemplars ($K = 500$ when using exemplars). We give the results of FeTrIL, EFC, and SS-IL alongside the accuracies initially reported by Magistri et al. [22]. We highlight the best-performing methods, both exemplar-free and exemplar-based, in **bold** and underline the second-best approaches.

Method	Accuracy ↑ 14 classes w/o exemplars		Accuracy ↑ 14 classes w/ exemplars	
	Patch	Image	Patch	Image
Finetuning (**LB**)	23.3 (± 6.4)	22.5 (± 3.3)	49.0 (± 5.2)	52.1 (± 7.8)
EWC [23]	27.3 (± 2.1)	25.3 (± 2.2)	53.3 (± 5.2)	56.0 (± 7.0)
LwF [25]	26.0 (± 6.7)	25.6 (± 7.3)	48.8 (± 5.9)	51.1 (± 8.0)
RWalk [24]	28.2 (± 8.4)	25.8 (± 7.5)	51.3 (± 5.4)	53.9 (± 7.4)
FeTrIL [31]	36.4 (± 3.4)	40.3 (± 5.8)	–	–
EFC [32]	**39.4** (± 3.7)	**46.6** (± 6.0)	–	–
BIC [26]	–	–	**56.4** (± 6.1)	**63.3** (± 3.6)
iCaRL [21]	–	–	54.7 (± 5.8)	62.5 (± 3.1)
IL2M [27]	–	–	49.4 (± 1.8)	52.0 (± 1.8)
SS-IL [29]	–	–	46.4 (± 8.3)	52.4 (± 5.2)
Joint Incremental (**UB**)	67.7 (± 4.8)	73.0 (± 4.2)	67.7 (± 4.8)	73.0 (± 4.2)

Table 5
Incremental Social Version Classification (ISVC) on 15 classes in TAG accuracy on patches and images after the last task with and without exemplars ($K = 500$ when using exemplars). We highlight the best-performing methods, both exemplar-free and exemplar-based, in **bold** and underline the second-best approaches.

Method	Accuracy ↑ 15 classes w/o exemplars		Accuracy ↑ 15 classes w/ exemplars	
	Patch	Image	Patch	Image
Finetuning (**LB**)	14.2 (± 3.7)	12.0 (± 3.9)	27.0 (± 1.5)	28.4 (± 2.7)
EWC [23]	20.1 (± 4.2)	14.4 (± 3.7)	22.3 (± 1.5)	25.8 (± 2.9)
LwF [25]	19.5 (± 0.4)	11.2 (± 0.5)	29.4 (± 2.2)	31.9 (± 3.1)
RWalk [24]	18.6 (± 1.5)	12.7 (± 1.6)	24.4 (± 4.6)	26.0 (± 3.2)
FeTrIL [31]	41.0 (± 0.6)	45.5 (± 0.7)	–	–
EFC [32]	**48.5** (± 2.1)	**51.9** (± 0.7)	–	–
BIC [26]	–	–	40.9 (± 3.5)	44.3 (± 3.5)
iCaRL [21]	–	–	40.9 (± 2.9)	**50.7** (± 2.3)
IL2M [27]	–	–	24.2 (± 4.4)	27.3 (± 4.4)
SS-IL [29]	–	–	**45.6** (± 1.3)	47.9 (± 2.0)
Joint Incremental (**UB**)	63.3 (± 1.5)	66.3 (± 0.9)	63.3 (± 1.5)	66.3 (± 0.9)

Table 6
Incremental Social Version Classification (ISVC) on 4 classes in average TAG accuracy on patches and images after the last task with and without exemplars ($K = 500$ when using exemplars). We highlight the best-performing methods, both exemplar-free and exemplar-based, in **bold** and underline the second-best approaches.

Method	Accuracy ↑ 4 classes w/o exemplars		Accuracy ↑ 4 classes w/ exemplars	
	Patch	Image	Patch	Image
Finetuning (**LB**)	39.6 (± 7.0)	43.0 (± 6.1)	53.4 (± 1.2)	57.4 (± 1.2)
EWC [23]	41.5 (± 8.4)	45.6 (± 10.1)	47.2 (± 2.1)	53.5 (± 2.6)
LwF [25]	39.3 (± 0.1)	45.2 (± 0.2)	56.3 (± 1.9)	61.3 (± 2.6)
RWalk [24]	35.3 (± 3.1)	37.2 (± 3.4)	49.7 (± 5.0)	54.3 (± 3.9)
FeTrIL [31]	69.3 (± 0.6)	72.9 (± 0.6)	–	–
EFC [32]	**71.4** (± 0.4)	**73.7** (± 0.8)	–	–
BIC [26]	–	–	**70.4** (± 1.8)	**72.2** (± 1.5)
iCaRL [21]	–	–	63.3 (± 1.1)	67.1 (± 1.0)
IL2M [27]	–	–	49.5 (± 6.8)	55.2 (± 5.6)
SS-IL [29]	–	–	66.5 (± 1.2)	67.6 (± 1.6)
Joint Incremental (**UB**)	85.9 (± 0.8)	87.8 (± 0.5)	85.9 (± 0.8)	87.8 (± 0.5)

ISPC scenario. Our conjecture is that this outcome can be attributed to the task-wise knowledge distillation of SS-IL, which could potentially necessitate a greater number of exemplar samples for effective learning in this context.

Results show that exemplar-based approaches are capable of reaching performance comparable to that of Joint Incremental in both the ISPC 14-class scenario and ISVC 15-class scenario while retaining only a fraction of the examples (see Tables 4 and 5). However, there is a larger performance gap observed in the ISVC 4-class scenario compared to the other scenarios. EFC and FeTrIL perform worse than exemplar-based approaches when the memory size increases, however they still achieve competitive performance in the ISVC 4-class scenario. Finally, it is worth noting that iCaRL performs exceptionally well when the memory is limited to storing only $K = 100$ image patches. This highlights the effectiveness of iCaRL even with a significantly reduced memory size.

6. Conclusions

In this paper, we extended the work of Magistri et al. [22] on the advantages of applying continual learning approaches to the task of social network identification. We considered two practical situations where updating an existing model would be valuable: Incremental

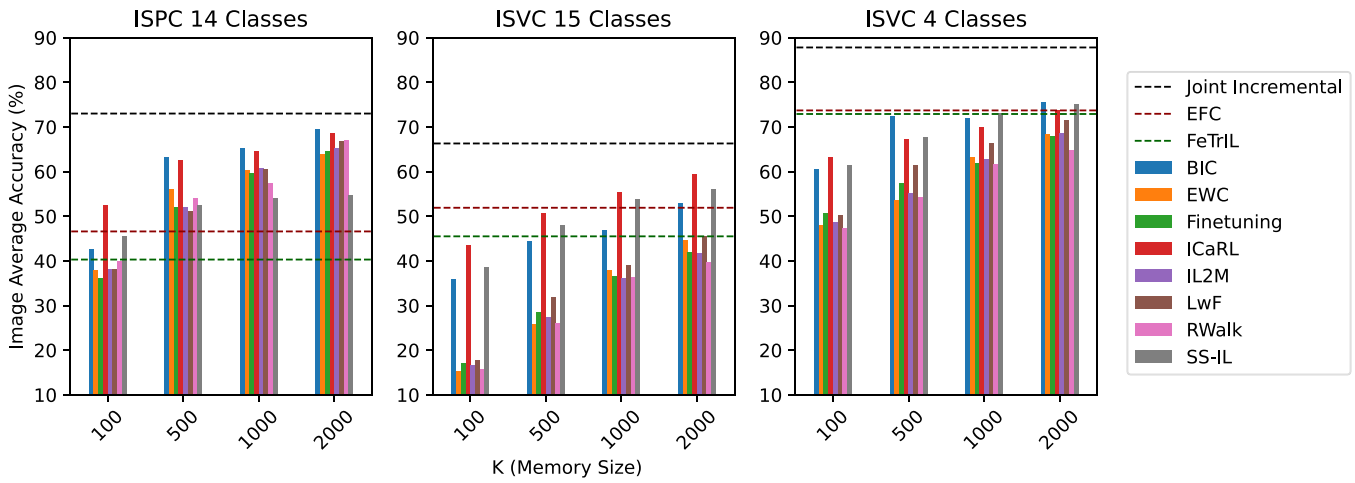

Fig. 3. Image average accuracy as a function of memory size in the ISPC and ISVC scenarios.

Social Platform Classification, which involves accommodating newly introduced platforms, and Incremental Social Version Classification, which entails handling updated versions of existing social networks. To evaluate the effectiveness of incremental updating, we conducted extensive experiments with exemplar-free and exemplar-based continual learning methods to incrementally update a state-of-the-art network. Remarkably, exemplar-free methods based on prototypes provide a viable solution when saving previous tasks exemplars is not feasible, for instance due to privacy concerns. Exemplars-based approaches achieve the largest improvement over finetuning in all considered scenarios by retaining only a fraction of the original training patches. Even though continual learning methods are not yet able to reach the performance obtained by Joint Incremental Training, the reported results shows that recent techniques are rapidly closing the gap with the upper bound. This extensive evaluation serves as an initial benchmark, providing a foundation for researchers to further explore continual social network identification in their studies.

As future work, exemplar-based methods could be further improved by employing a patch selection strategy based on the distribution of DCT coefficients. Moreover, considering the large gap between joint incremental and continual learning approaches in the ISVC scenario, we hypothesize that there exist features shared across different tasks that are not currently taken into account by exemplar-based methods. To address this, future research efforts could concentrate on expanding continual learning approaches to identify and incorporate these inter-task features.

CRediT authorship contribution statement

Simone Magistri: Conceptualization, Data curation, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Daniele Baracchi:** Investigation, Methodology, Writing – original draft, Writing – review & editing. **Dasara Shullani:** Data curation, Investigation, Writing – original draft, Writing – review & editing. **Andrew D. Bagdanov:** Conceptualization, Investigation, Supervision, Writing – original draft, Writing – review & editing, Resources. **Alessandro Piva:** Conceptualization, Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Alessandro Piva reports financial support was provided by Defense Advanced Research Projects Agency. Alessandro Piva reports financial support was provided by Italian Ministry of Universities and Research.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the Italian Ministry of Universities and Research (MUR) under Grant 2017Z595XS, and in part by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00112090136. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

References

[1] C. Pasquini, I. Amerini, G. Boato, Media forensics on social media platforms: a survey, EURASIP J. Inf. Secur. 2021 (1) (2021) 1–19.

[2] P. Bestagini, M. Fontani, S. Milani, M. Barni, A. Piva, M. Tagliasacchi, S. Tubaro, An overview on video forensics, in: European Signal Processing Conference, 2012, pp. 1229–1233.

[3] A. Castiglione, G. Cattaneo, A. De Santis, A forensic analysis of images on online social networks, in: International Conference on Intelligent Networking and Collaborative Systems, IEEE, 2011, pp. 679–684.

[4] O. Giudice, A. Paratore, M. Moltisanti, S. Battiato, A classification engine for image ballistics of social data, in: International Conference on Image Analysis and Processing, Springer, 2017, pp. 625–636.

[5] J. He, Z. Lin, L. Wang, X. Tang, Detecting doctored JPEG images via DCT coefficient analysis, in: European Conference on Computer Vision, Springer, 2006, pp. 423–435.

[6] T. Pevny, J. Fridrich, Detection of double-compression in JPEG images for applications in steganography, IEEE Trans. Inf. Forensics Secur. 3 (2) (2008) 247–258.

[7] I. Amerini, T. Uricchio, R. Caldelli, Tracing images back to their social network of origin: A CNN-based approach, in: IEEE Workshop on Information Forensics and Security, IEEE, 2017, pp. 1–6.

[8] D. Shullani, D. Baracchi, M. Iuliani, A. Piva, Social network identification of laundered videos based on dct coefficient analysis, IEEE Signal Process. Lett. 29 (2022) 1112–1116.

[9] Manisha, A. Karunakar, C.-T. Li, Identification of source social network of digital images using deep neural network, Pattern Recognit. Lett. (ISSN: 0167-8655) 150 (2021) 17–25.

[10] J. Lukas, J. Fridrich, M. Goljan, Digital camera identification from sensor pattern noise, IEEE Trans. Inf. Forensics Secur. 1 (2) (2006) 205–214.

[11] A. Castiglione, G. Cattaneo, M. Cembalo, U. Ferraro Petrillo, Experimentations with source camera identification and online social networks, J. Ambient Intell. Humaniz. Comput. 4 (2) (2013) 265–274.

[12] R. Caldelli, I. Amerini, C.T. Li, PRNU-based image classification of origin social network with CNN, in: 26th European Signal Processing Conference, IEEE, 2018, pp. 1357–1361.
[13] I. Amerini, C.-T. Li, R. Caldelli, Social network identification through image classification with CNN, IEEE Access 7 (2019) 35264–35273.
[14] T. Gloe, Forensic analysis of ordered data structures on the example of jpeg files, in: IEEE International Workshop on Information Forensics and Security, IEEE, 2012, pp. 139–144.
[15] S. Verde, C. Pasquini, F. Lago, A. Goller, F. De Natale, A. Piva, G. Boato, Multi-clue reconstruction of sharing chains for social media images, IEEE Trans. Multimed. 25 (2023) 9491–9505.
[16] M. Iuliani, D. Shullani, M. Fontani, S. Meucci, A. Piva, A video forensic framework for the unsupervised analysis of MP4-like file container, IEEE Trans. Inf. Forensics Secur. 14 (3) (2018) 635–645.
[17] P. Yang, D. Baracchi, M. Iuliani, D. Shullani, R. Ni, Y. Zhao, A. Piva, Efficient video integrity analysis through container characterization, IEEE J. Sel. Top. Sign. Proces. 14 (5) (2020) 947–954.
[18] K. Rana, G. Singh, P. Goyal, SNRCN2: Steganalysis noise residuals based CNN for source social network identification of digital images, Pattern Recognit. Lett. 171 (2023) 124–130.
[19] M. McCloskey, N.J. Cohen, Catastrophic interference in connectionist networks: The sequential learning problem, in: Psychology of Learning and Motivation, vol. 24, Elsevier, 1989, pp. 109–165.
[20] F. Marra, C. Saltori, G. Boato, L. Verdoliva, Incremental learning for the detection and classification of gan-generated images, in: IEEE International Workshop on Information Forensics and Security, IEEE, 2019, pp. 1–6.
[21] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C.H. Lampert, Icarl: Incremental classifier and representation learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2001–2010.
[22] S. Magistri, D. Baracchi, D. Shullani, A.D. Bagdanov, A. Piva, Towards continual social network identification, in: 11th International Workshop on Biometrics and Forensics, 2023, pp. 1–6.
[23] 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, Proc. Natl. Acad. Sci. 114 (13) (2017) 3521–3526.
[24] A. Chaudhry, P.K. Dokania, T. Ajanthan, P.H.S. Torr, Riemannian walk for incremental learning: Understanding forgetting and intransigence, in: Proceedings of the European Conference on Computer Vision, 2018.
[25] Z. Li, D. Hoiem, Learning without forgetting, IEEE Trans. Pattern Anal. Mach. Intell. 40 (12) (2017) 2935–2947.
[26] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, Y. Fu, Large scale incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[27] E. Belouadah, A. Popescu, IL2m: Class incremental learning with dual memory, in: IEEE/CVF International Conference on Computer Vision, 2019, pp. 583–592.
[28] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, 2015, URL https://arxiv.org/abs/1503.02531.
[29] H. Ahn, J. Kwak, S.F. Lim, H. Bang, H. Kim, T. Moon, SS-il: Separated softmax for incremental learning, IEEE/CVF Int. Conf. Comput. Vis. (2020) 824–833.
[30] M. Masana, X. Liu, B. Twardowski, M. Menta, A.D. Bagdanov, J. van de Weijer, Class-incremental learning: survey and performance evaluation on image classification, IEEE Trans. Pattern Anal. Mach. Intell. (2022).
[31] G. Petit, A.-S. Popescu, H. Schindler, D. Picard, B. Delezoide, Fetril: Feature translation for exemplar-free class-incremental learning, IEEE/CVF Winter Conf. Appl. Comput. Vis. (2022) 3900–3909.
[32] S. Magistri, T. Trinci, A. Soutif, J. van de Weijer, A.D. Bagdanov, Elastic feature consolidation for cold start exemplar-free incremental learning, in: The Twelfth International Conference on Learning Representations, 2024, https://openreview.net/forum?id=7D9X2cFnt1.
[33] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[34] SmartData Research Group, Smartphone images dataset, 2015, URL http://smartdata.cs.unibo.it/datasets.
[35] B. Hadwiger, C. Riess, The forchheim image database for camera identification in the wild, in: Pattern Recognition. ICPR International Workshops and Challenges, Springer, 2021, pp. 500–515.
[36] D. Baracchi, D. Shullani, M. Iuliani, D. Giani, A. Piva, Uncovering the authorship: Linking media content to social user profiles, Pattern Recognition Letters (2024) submitted.
[37] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, URL https://arxiv.org/abs/1412.6980.
[38] M. Kivanc Mihcak, I. Kozintsev, K. Ramchandran, Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising, in: IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings., Vol. 6, 1999, pp. 3253–3256 vol.6.