

Continuous fake media detection: adapting deepfake detectors to new generative techniques

Francesco Tassone

tassone.1814263@studenti.uniroma1.it
Sapienza University of Rome
Rome, Rome, Italy

Luca Maiano

maiano@diag.uniroma1.it
Sapienza University of Rome
Rome, Rome, Italy
Ubiquitous srl
Rome, Rome, Italy

Irene Amerini

amerini@diag.uniroma1.it
Sapienza University of Rome
Rome, Rome, Italy

ABSTRACT

Generative techniques continue to evolve at an impressively high rate, driven by the hype about these technologies. This rapid advancement severely limits the application of deepfake detectors, which, despite numerous efforts by the scientific community, struggle to achieve sufficiently robust performance against the ever-changing content. To address these limitations, in this paper, we propose an analysis of two continuous learning techniques on a *Short* and a *Long* sequence of fake media. Both sequences include a complex and heterogeneous range of deepfakes (generated images and videos) from GANs, computer graphics techniques, and unknown sources.

Our experiments show that continual learning could be important in mitigating the need for generalizability. In fact, we show that, although with some limitations, continual learning methods help to maintain good performance across the entire training sequence. For these techniques to work in a sufficiently robust way, however, it is necessary that the tasks in the sequence share similarities. In fact, according to our experiments, the order and similarity of the tasks can affect the performance of the models over time. To address this problem, we show that it is possible to group tasks based on their similarity. This small measure allows for a significant improvement even in longer sequences. This result suggests that continual techniques can be combined with the most promising detection methods, allowing them to catch up with the latest generative techniques.

In addition to this, we propose an overview of how this learning approach can be integrated into a deepfake detection pipeline for continuous integration and continuous deployment (CI/CD). This allows you to keep track of different funds, such as social networks, new generative tools, or third-party datasets, and through the integration of continuous learning, allows constant maintenance of the detectors.

KEYWORDS

Continual learning, deepfake detection, MLOps, CI/CD

1 INTRODUCTION

Generative AI tools like Midjourney¹, ChatGPT², or the more recent Sora³ are completely revolutionizing the way media content is created, leading to the mass adoption of tools that were unimaginable

until recently. However, this progress makes the threat of new-generation disinformation or defamation campaigns increasingly concrete. Unfortunately, forensic tools for detecting this content are not advancing at the same rate. Although it is possible to train highly accurate detectors, these methodologies still poorly generalize to new generative methods due to data drift [19]. Detectors perform well on the generative techniques they are trained on but commonly fail when exposed to content generated with a new generative model.

Because of these limitations, the practical application of automatic detectors has been almost nil. When someone wants to deploy these tools in commercial or mass verification systems, they must face numerous challenges that go far beyond the need to generalize from a few known benchmarks [4, 16, 24, 32]. Prominent among these is the need to continuously train these models on new generative techniques through continuous learning methods. To address this continuous change, we analyze continuous learning techniques to evaluate their limitations when applied to mitigate this problem. Continual learning, also known as lifelong learning or incremental learning, is a constant learning approach. Unlike transfer learning techniques, where a model trained on one task is retrained on a new task to improve its performance on the latter, continual learning involves maintaining good model performance on a set of evolving tasks as these become available without incurring in *catastrophic forgetting* [8]. These requirements fit well with the problem of learning to recognize content generated by new techniques and the need to continually readjust a model with respect to data drift produced by the shift in the distribution of data observed in inference versus those seen in training.

This work aims to investigate the effectiveness of two continuous learning techniques with the intention of integrating them into a real, end-to-end deepfake detection system that allows for continuous integration and continuous delivery/deployment (CI/CD). Our goal is therefore to propose a simple yet effective design for a Machine Learning Model Operations (*MLOps* [19, 25]) pipeline that would enable the end-to-end development of continuously trained and monitored intelligent detectors with a minimal set of components.

Contributions Therefore, the main contributions of this preliminary study are summarized below.

- *Analysis of continual learning methods.* We study the effectiveness of two continuous learning methods, *Knowledge Distillation* [10] (KD) and *Elastic Weight Consolidation* [14] (EWC), and show their superiority to transfer learning when continuous training is needed.

¹<https://www.midjourney.com/>

²<https://openai.com/chatgpt>

³<https://openai.com/sora>

- *Sequence*. We study how the order of arrival of the tasks can affect the performance of the model. In particular, we show how task similarity plays an important role in maintaining optimal performance.
- *Multi-task continual training*. We show how aggregating tasks based on their similarity can significantly improve the overall performance over the entire sequence. This result is particularly important and helps us to better outline the possible developments of these techniques for deepfake recognition.
- *CI/CD pipeline for deepfake detection*. We propose an overview of an end-to-end system for continuous integration and continuous delivery/deployment for a deepfake detection application.

The rest of this paper is organized as follows. Section 2 offers an overview of the state of the art. In Section 3, we introduce the methodology used in this study. In Section 4, we present the experiments we conducted. Finally, in Section 5, we draw the final considerations and illustrate the future developments of this work.

2 RELATED WORKS

Despite the difficulties in keeping up with the advancement of generative techniques, numerous studies have been proposed on detecting generated images and videos [2, 26]. In this section, we provide an overview of the most recent advances. However, it is essential to note that many can be combined with the continuous learning techniques analyzed in this paper. In fact, this type of learning approach could be used not as an alternative to these methods but to make these techniques maintainable over time.

First of all, several studies have found that there are some key ingredients for more robust detection [9, 28]. Among them, image compression and resizing can severely mitigate model performance [21]. Therefore, to cope with these problems, it is usually recommended to avoid resizes, as they entail image resampling and interpolation, which may erase the subtle high-frequency traces left by the generation process and train models with different forms of augmentation. Moreover, working on local patches also appears to be important [3] as well as analyzing both local and global features [11].

A recent study from Aghasani et al. [1] showed that foundational models like ViT can effectively distinguish between authentic and counterfeit images, even when interpretability through prototypes is important. Additionally, the study demonstrated that classifiers with fine-tuned features consistently outperform those utilizing pre-trained weights when applied to cross-dataset domains. Another study by Le et al. [15] considers quality factors to train robust detectors. The authors used an intra-model collaborative learning method to minimize the geometrical differences of images in various qualities at different intermediate layers. This idea, combined with an adversarial weight perturbation module, can be used to improve the robustness of the model against input image compression. Many recent studies also focus on combining different modes, such as audio and video [22, 30] or video and depth [17], as well as open-set recognition [27].

Other studies focus on reconstructing fake artifacts introduced by generative models by considering second-order statistics in the

spatial and frequency domains [28]. Corvi et al. [4] showed that, similar to GANs, diffusion models also give rise to visible artifacts in the Fourier domain and exhibit anomalous regular patterns in autocorrelation. In fact, synthetic and real images exhibit significant differences in the mid-high frequency signal content, observable in their radial and angular spectral power distributions.

Among the various detection strategies, watermarks have also been proposed [7, 29]. Through the addition of special information within the image being generated, these watermarks can be used to verify the generative model used to create content. For example, Zhao et al. [31] proposed an encoder-decoder network to embed watermarks as anti-deepfake labels into the facial identity features. The injected label is entangled with the facial identity feature, so it will be sensitive to face swap translations and robust to conventional image modifications like resizing and compression. However, these solutions are limited in that they require a model to integrate the watermark into the content during the generation phase.

Our method, instead, falls into a different line of studies targeted at designing continual learning methods for deepfake detection. This learning approach has only been partially explored for this task. The study by Marra et al. [18] was one of the first to propose a multi-task incremental learning method for GAN-generated images based on iCaRL [23]. A similar approach was applied to videos in Khan et al. [12], while Pan et al. [20] have recently proposed to learn semantically consistent representations across domains based on supervised contrastive learning and a carefully designed replay set. In contrast, Kim et al. [13] combined a knowledge distillation method with a representation learning loss. However, these methods have been tested on datasets not explicitly designed for this type of learning approach, which very often consists of a limited number of generative techniques. To overcome this limitation, Li et al. [16] recently introduced a new collection of deepfakes from known and unknown generative models. The proposed CDDB dataset includes multiple evaluations on detecting an easy, hard, and long sequence of deepfakes with appropriate measures. For these reasons, in this paper, we focus our studies on this dataset by measuring the performance of the continual learning methods examined in this paper on this dataset. Different from other studies, we analyze this specific learning method to integrate it into a CI/CD system appropriately designed for this task in the future.

3 METHOD

The practical application of deepfake detectors has been severely limited by the need to develop robust detectors with respect to new generative techniques. To overcome this problem, we propose an analysis of two continual learning strategies and an overview of a simple CI/CD pipeline that can complement this preliminary study. This pipeline (depicted in Figure 1) has the advantage of being constantly updated with respect to the latest detectors, which can be adapted for continuous learning as described below.

3.1 Learning strategies

To evaluate the effectiveness of continual learning in the deepfake recognition task we examine two learning methods. These techniques were chosen based on their demonstrated effectiveness in other tasks [5].

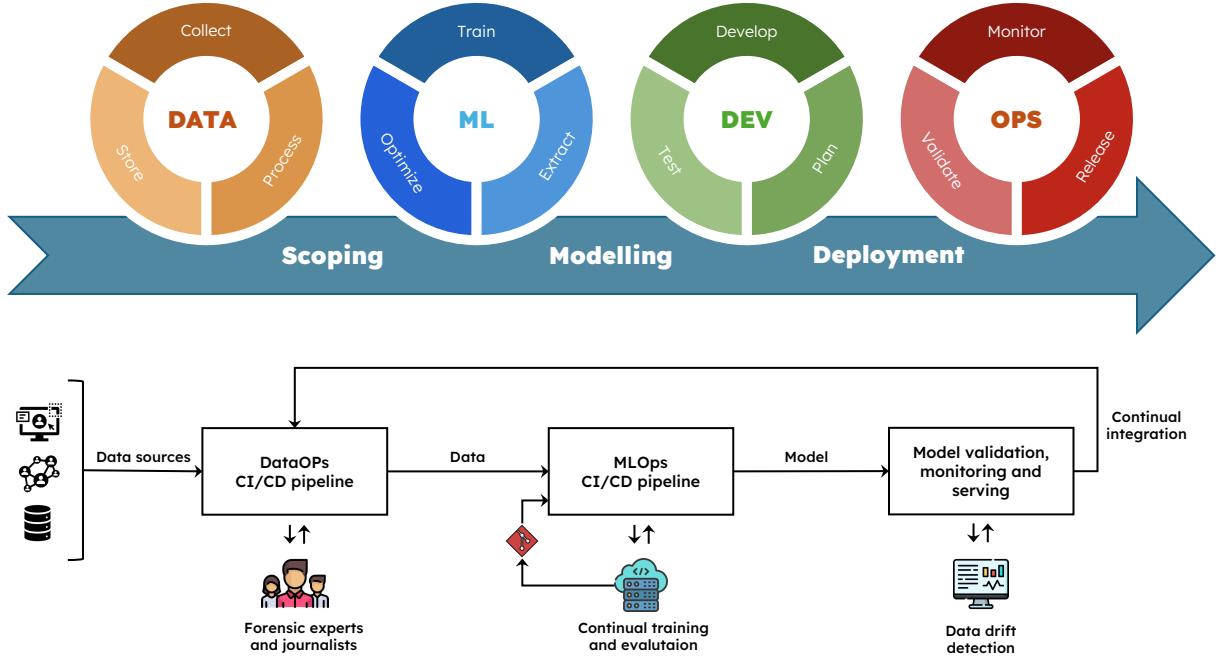


Figure 1: Proposed CI/CD pipeline for deepfake detection. The data coming from different sources like new generative tools, social media or existing databases are analyzed by forensic experts for continual retraining of the system. Next, these data are used for continual learning and monitoring. The data drift distribution module rise an alert whenever it detects new input data distributions. The continual learning methods analyzed in this paper are part of the *MLOps CI/CD pipeline* block in the figure.

Knowledge Distillation (KD). Distillation techniques were introduced by Hinton et al. [10] in order to transfer knowledge from a neural network \mathcal{T} (the *teacher*) to a neural network \mathcal{S} (the *student*). The key idea behind knowledge distillation is that soft probabilities predicted by a network of trained "teachers" contain much more information about a data point than a simple class label. For example, if multiple classes are assigned high probabilities for an image, this could mean that the image must be close to a decision boundary between those classes. Forcing a student to mimic these probabilities should then cause the student network to absorb some of this knowledge that the teacher discovered above and beyond the information in training labels alone. To implement the KD strategy, we modify the cross-entropy loss \mathcal{L}_S by adding a regularization term (\mathcal{L}_D) as follows.

$$\mathcal{L}_{KD}(\theta) = \alpha \mathcal{L}_D + \beta \mathcal{L}_S \quad (1)$$

Here, α and β are scalar coefficients that control the balancing between the current and past tasks. The distillation loss \mathcal{L}_D uses the output of the teacher model to facilitate knowledge transfer from previous tasks to the new student model. As the teacher is non-trainable, its predictions are solely based on prior knowledge, producing soft labels for the training process of the student model. This term is computed as:

$$\mathcal{L}_D = \sum_{x_i \in X} \sigma(\mathcal{T}(x_i, \hat{y}_i), \tau) \log \sigma(\mathcal{S}(x_i, y_i), \tau) \quad (2)$$

where \hat{y} represents the output of \mathcal{T} , σ denotes the softmax function with temperature τ and \mathcal{S} represent the student network.

Elastic Weight Consolidation (EWC). EWC remembers old tasks by selectively slowing down learning on weights that are important for these tasks. As shown in Kirkpatrick et al. [14], learning from a task A to a task B , there exist many configurations of θ leading to the same performance. In fact, the over-parametrization of the model makes it more likely the existence of a solution θ_B^* for task B that is close to task A . Therefore, previous tasks' performances are kept by constraining, with a quadratic penalty, the parameters to stay in a region centered in θ_A^* of low error for task A . Formally, the function \mathcal{L} that we minimize in EWC is:

$$\mathcal{L}_{EWC}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i (\theta_i - \theta_{A,i}^*)^2 \quad (3)$$

where F is the Fisher information matrix, λ sets how important the old task is compared to the new one and i labels each parameter. When moving to a third task (i.e., task C), EWC will try to keep the network parameters close to the learned parameters of both task A and B . This can be enforced either with two separate penalties, or as one by noting that the sum of two quadratic penalties is itself a quadratic penalty.

3.2 Training procedure

The training procedure for both learning methods described in the previous section is summarized in Algorithm 1. Given a stream of

AI-generated contents $\mathcal{D} = \{D_1, \dots, D_n\}$, $n \geq 1$, at each training iteration t we train a model $g_t(x_t, \theta_t)$ on the actual $\mathcal{D}_t \in \mathcal{D}$. After the first training iteration on the first available batch of samples $\mathcal{D}_1 \in \mathcal{D}$, the model is trained on all next batches $\{\mathcal{D}_2, \dots, \mathcal{D}_n\} \in \mathcal{D}$ for all $n > 1$. Differently from transfer learning, however, the model is forced to minimize the loss function for both new and old examples without requiring training over the past data samples, therefore learning an optimal set of parameters θ_t for all observed inputs $(x_i, y_i), i \in \{0, \dots, t\}$.

Algorithm 1 Training procedure for continual learning methods.

```

Require:  $\mathcal{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_n\}, n \geq 1$ 
 $\mathcal{D}_t = \mathcal{D}.pop()$ 
 $g_t(x_t, \theta_t) \leftarrow train(\mathcal{D}_t)$ 
while  $\mathcal{D}.isEmpty()$  do
     $\mathcal{D}_t = \mathcal{D}.pop()$ 
    if strategy == KD then
         $\mathcal{T} \leftarrow g_t(x_t, \theta_t).copy()$ 
         $\mathcal{S} \leftarrow train(\mathcal{T}, \mathcal{D}_t)$   $\triangleright$  Train using the  $\mathcal{L}_{KD}$  loss in Eq. 1.
         $g_t(x_t, \theta_t) \leftarrow \mathcal{S}$ 
    else if strategy == EWC then
         $g_t(x_t, \theta_t) \leftarrow train(\mathcal{D}_t)$   $\triangleright$  Train using the  $\mathcal{L}_{EWC}$  loss in
        Eq. 3.
    end if
end while

```

3.3 Lightweight CI/CD for deepfake detection

We conclude this section by providing an overview of how the proposed methodology could be integrated into a continuous integration and continuous delivery pipeline. The Figure 1 shows the complete pipeline. The upper part of the figure shows all the phases of an MLOps CI/CD system. The lower part shows the complete pipeline into which the continuous learning methods analyzed in this paper can be integrated.

The pipeline consists of three main modules. In the first, data from generative tools, social media or databases are organized and analyzed if necessary by forensic experts or journalists. This module enables the preparation of model training data. In the next module, the model continuously learns from incoming data. This module also allows you to keep several copies of the model in case you need to restore a previous version. Finally, in the last part, a continuous delivery and monitoring module takes care of serving the newly trained model to check for any data drift. If a new distribution of input data is identified (e.g., content generated with a new technique), it is immediately saved and flagged so that the model can be verified and possibly retrained on the new data.

4 EXPERIMENTS

In this section we discuss the analyses conducted to evaluate the effectiveness of the continual learning methods introduced in Section 3.1. We begin by introducing the dataset and selected architectures in Section 4.1, and then, in Section 4.2 we analyze the performance of the two continual learning strategies by comparing them with transfer learning.

4.1 Experimental setting

Dataset. We use the CDDB [16] dataset for all our experiments. The dataset offers three different evaluation scenarios: an easy task sequence, a hard task sequence, and a long one. The dataset contains media generated with 5 different types of GAN-based generative models (StyleGAN, BigGAN, CycleGAN, GauGAN, and StarGAN), 5 non-GAN models (Glow, CRN, IMLE, SAN, and FaceForensics++), and two datasets whose origin is unknown (WhichFaceIsReal and WildDeepfake).

For this study, we report the results on the easy and long sequences. We refer to these two sequences as *Easy* and *Long*, respectively.

- The *Easy* setup (composed of GauGAN, BigGAN, CycleGAN, IMLE, FaceForensics++, CRN, and WildDeepfake) is used to study the basic behavior of evaluated methods when they address similar generative techniques.
- The *Long* setup (composed of GauGAN, BigGAN, CycleGAN, IMLE, FaceForensics++, CRN, WildDeepfake, Glow, StarGAN, StyleGAN, WhichFaceReal, and SAN) is designed to encourage methods to better handle long sequences of deepfake detection tasks, where the catastrophic forgetting might become more serious.

Architecture. For our analysis, we selected four different state-of-the-art that have demonstrated to achieve good results on this task [9, 21]: Resnet-50, Resnet-18, Mobilenet-V2, and Xception.

4.2 Analysis

We now turn to analyze the performance of continuous learning methods with selected backbones. We start by illustrating the results on the *short* set and then extend the considerations to the longer (i.e., the *long* set). We trained each model using early stopping with a patience of 35 up to 250 epochs. We used the Stochastic Gradient Descent (SGD) optimizer to modify the models' weights, with an initial learning rate of 0.005 and a momentum of 0.1. The learning rate is controlled by a cosine annealing scheduler with a minimum value of 10^{-5} . For preprocessing, we applied a random crop on each image with a resolution of 128×128 .

Zero shot. We begin by analyzing the results of all models trained on the GauGAN task and evaluating the whole *long* sequence. As shown in Figure 2, all the models almost always fail to detect tasks outside their training data, indicating a clear lack of generalizability. These evaluations confirm that a model trained on a particular generative technique struggles to detect other types of fake images. In fact, we can see that the model manages to achieve more or less satisfactory performances on media generated with GANs (in particular BigGAN, CycleGAN, and StarGAN), which evidently have characteristics more similar to those seen in the training phase, but it ultimately fails the tasks more complex ones like FaceForensics++ or WildDeepfake.

Short set. In Table 1, we report the experiments conducted on this set. The table reports the performance of each dataset trained on the entire sequence. For example, Resnet-50 obtains an accuracy of 57.80% on GauGAN after being trained with KD up to the last

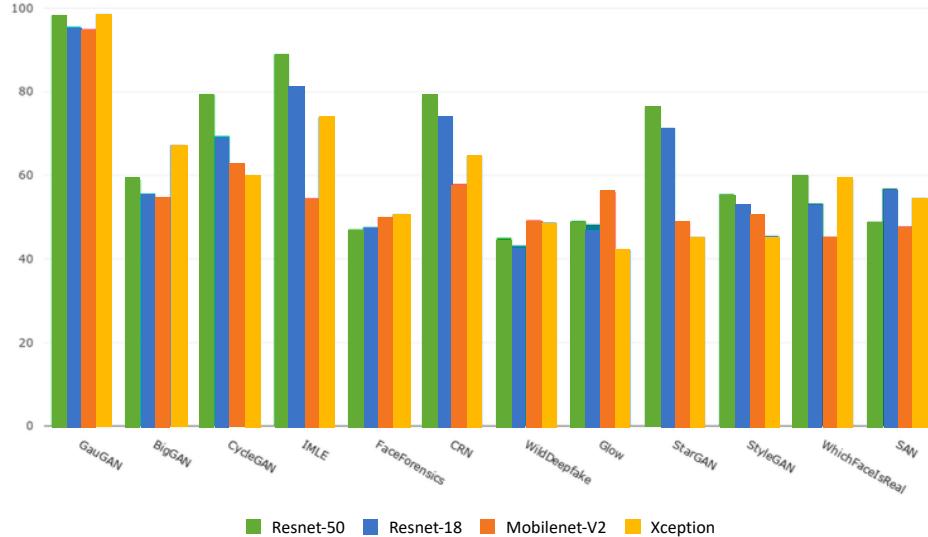


Figure 2: Zero-shot performance on the *Long* set. All the models are trained only on the GauGAN task and evaluated over the whole dataset.

available dataset in the sequence (i.e., WildDeepfake). From the table, we can draw some initial insights. Starting with the backbones, the Resnet-50 and Mobilenet-V2 achieve the best results on average. As for learning techniques, we can see a general improvement in the performance of continual learning methods compared to transfer learning. In particular, Knowledge Distillation achieves the best performance when combined with Resnet-50. In this specific configuration, the model achieves excellent performance on IMLE (97.30%) and CRN (94.36%), and good performance on CycleGAN (79.85%), while the results on the other datasets range between 51.22% and 64.45%. The results are not surprisingly high, but to understand why, it is necessary to reason about the different types of datasets contained in this sequence. IMLE and CRN both contain media drawn from computer games, which are indeed more easily recognized. GauGAN, BigGAN, and CycleGAN are all datasets generated with Generative Adversarial Networks, so they have similar characteristics. Finally, FaceForensics++ and WildDeepfake are both challenging sets containing diverse generative techniques. These characteristics make the sequence highly varied and therefore complex to be classified uniformly well. In particular, as we iterated through the various tasks in the sequence, we noticed that performance can fluctuate significantly depending on the order in which these datasets are used.

In Figures 3 and 4, we elaborate more on this behavior. The figure shows the average performance on each dataset recorded by training the model on the various tasks in the following order: (1) GauGAN, (2) BigGAN, (3) CycleGAN, (4) IMLE, (5) FaceForensics++, (6) CRN, and (7) WildDeepfake. From the figure, we can see that the performance remains high on average on the first three (GAN-based) tasks and then undergoes an initial slight decrease with IMLE and stabilizes with a more substantial decrease from the fifth task onward. By analyzing this behavior, we can infer that the big difference between the generative techniques present in

FaceForensics++ and the previous tasks strains the model in finding a region of optimum that reduces the error on all tasks. This result is further confirmed with the arrival of WildDeepfake, which turns out to be the most complex and different task overall than the previous ones. The strong *asymmetry* of some tasks compared to others seems to play an important role, leading the model to optimize performance towards some families of tasks rather than others.

To confirm this hypothesis we tried to repeat the experiment by removing WildDeepfake from the sequence. As we can see from Figure 5, the performance of the models improves significantly if we compare the performance of the model trained on the complete sequence (Figure 5a) compared to the one trained on the sequence without WildDeepfake (Figure 5b). This tells us that the symmetry between the different tasks is fundamental to maintaining overall satisfactory performance on the entire sequence.

Overall, the results suggest a few things. First, we can notice a greater stability of continual learning techniques compared to transfer learning. Furthermore, Knowledge Distillation appears to be more robust overall than EWC, achieving higher performance on average than its rival. In all cases, all techniques achieve better results than the zero-shot scenario. However, the order of the sequence and the similarity of the tasks seem to play a significant role in overall performance. Finally, the most robust model seems to be the Resnet-50, followed closely by the Mobilenet-V2.

Long set. To complete the analysis, we extend the considerations made up to now to the *long* sequence. In Figure 6, we report the average accuracy values on each task. As for the *short* sequence, the heterogeneity of the 12 tasks poses a challenge to the continual learning methods, preventing them from learning a general representation applicable to all tasks. In particular, the similar result

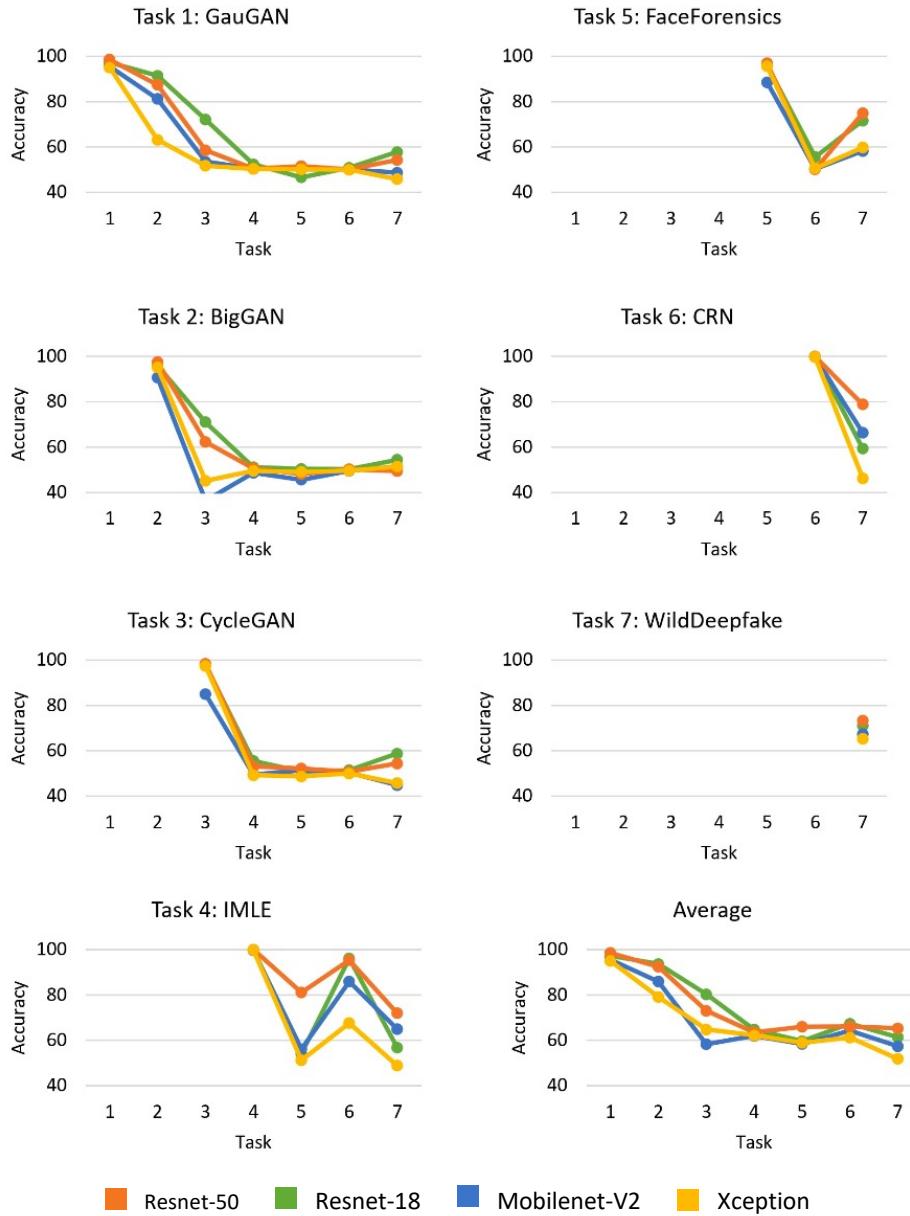


Figure 3: Elastic Weight Consolidation average accuracy at each task t calculated over tasks $\{1, \dots, t\}$. of all backbones on the Easy set. The order of the tasks is the following: (1) GauGAN, (2) BigGAN, (3) CycleGAN, (4) IMLE, (5) FaceForensics++, (6) CRN, and (7) WildDeepfake.

between the continual learning methods and transfer learning suggests that the latter cannot mitigate *catastrophic forgetting*. In fact, we can note that the models are able to achieve good performance on some tasks like IMLE and CRN (which have similar characteristics) and GAN-based tasks like GauGAN, BigGAN, StarGAN, and StyleGAN but fail to maintain acceptable performance on the other tasks.

Multi-task sequences. Collecting the considerations made so far, in this last test, we test the methods on a hybrid continuous learning scenario. Instead of learning each individual task separately, we combined the *Long* sequence tasks into groups of three, which we call *Multi-tasks*. Therefore, in this configuration, the sequence is composed of 4 macro tasks: $t_1 = \{\text{GauGAN}, \text{BigGAN}, \text{CycleGAN}\}$, $t_2 = \{\text{IMLE}, \text{FaceForensics}, \text{CRN}\}$, $t_3 = \{\text{WildDeepfake}, \text{Glow}, \text{StarGAN}\}$, and $t_4 = \{\text{StyleGAN}, \text{WhichFaceReal}, \text{SAN}\}$. Figure 7



Figure 4: Knowledge distillation average accuracy at each task t calculated over tasks $\{1, \dots, t\}$ of all backbones on the *Easy* set. The order of the tasks is the following: (1) GauGAN, (2) BigGAN, (3) CycleGAN, (4) IMLE, (5) FaceForensic++, (6) CRN, and (7) WildDeepfake.

reports the average accuracy values on each task. From the figure, we can immediately see that, as suggested in the previous experiment, WildDeepfake puts a strain on all learning techniques. It becomes clear that including images from unknown sources and tasks with a restricted dataset causes a drastic drop in accuracy. Moreover, regarding the learning techniques, all methods achieved satisfactory results. In particular, Knowledge Distillation confirms

itself as more robust in most of the sequence but seems to suffer a slight decline in the last three tasks. This result confirms that by aggregating tasks on the basis of their similarity, continuous learning techniques are able to maintain good performance over time.

Method	Model	Type							Average
		GauGAN	BigGAN	CycleGAN	IMLE	FaceForensics++	CRN	WildDeepfake	
Transfer learning	ResNet-50	57.80	54.38	58.79	<u>56.89</u>	71.63	59.32	<u>71.02</u>	61.40
	ResNet-18	48.70	51.50	44.87	65.02	58.23	<u>66.25</u>	67.32	57.41
	Mobilenet-V2	<u>54.30</u>	<u>49.38</u>	<u>54.40</u>	72.03	74.88	78.82	73.31	65.30
	Xception	45.90	51.38	45.79	48.87	59.81	46.16	65.26	51.88
EWC	ResNet-50	65.30	61.13	<u>62.64</u>	68.11	<u>73.77</u>	<u>71.77</u>	70.24	67.56
	ResNet-18	54.15	48.50	58.61	<u>72.50</u>	73.21	69.07	71.36	63.91
	Mobilenet-V2	<u>54.75</u>	<u>48.88</u>	63.37	76.96	86.98	89.62	<u>68.03</u>	69.80
	Xception	50.00	48.75	50.00	56.85	59.72	58.30	61.06	54.95
KD	ResNet-50	64.45	58.75	79.85	<u>97.30</u>	<u>64.65</u>	<u>94.36</u>	51.22	72.94
	ResNet-18	50.50	49.13	52.75	52.27	52.28	53.45	49.87	51.46
	Mobilenet-V2	<u>54.40</u>	<u>53.00</u>	<u>54.21</u>	80.28	72.19	86.37	<u>71.36</u>	<u>67.40</u>
	Xception	50.75	50.75	50.55	97.73	49.86	96.28	49.57	63.64

Table 1: Transfer learning, EWC, and KD performance on the *easy* set. The average column represents the average performance of each model across all datasets. Bold values represent the highest value for each learning method.

4.3 Limitations

The analyzed results give us some interesting information. First of all, continuous learning techniques show an improvement compared to the zero-shot scenario. However, the similarity of the various tasks in sequence and their order of arrival seem to play an important role in maintaining satisfactory performance throughout the sequence.

The results presented require further investigation. First of all, in this study, we limited ourselves to analyzing the behavior of 4 backbones commonly used in deepfake detection; however, a comparison of more advanced deepfake detection methods present in the literature is necessary. Furthermore, the results obtained in the multi-task configuration suggest that these techniques could benefit from using a memory [6]. This aspect will be further analyzed in our future studies. Finally, in this study, we assumed the different tasks arrived in batches. This is certainly possible, as the release of a new tool could lead to introducing a significant sample of media generated with it. However, it remains essential to study the robustness of these techniques on smaller and more heterogeneous sequences.

5 CONCLUSION

In this paper, we proposed an analysis of two deepfake detection techniques (Knowledge Distillation and Elastic Weight Consolidation), comparing them with zero-shot scenarios and transfer learning. The results show that continuous learning techniques can help make deepfake detection models more robust and easily updated to new generative methods. However, our analysis also highlighted some problems. The performance of these learning strategies seems to depend significantly on the similarity of the tasks and their order of arrival. To address this problem, we have shown how it is possible to combine the different tasks to obtain significantly better performance. Consequently, future developments of this work could analyze the importance of using memory and the robustness of these learning techniques to smaller and more heterogeneous sequences.

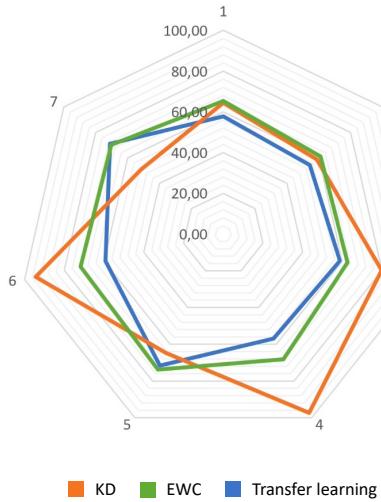
In addition to this, we also gave an overview of a CI/CD pipeline for deepfake detection, showing how the models used in this work can be combined with other modules to obtain a pipeline that can be used in a real application scenario. In this regard, in the future we will present a complete version of all the fundamental modules of this pipeline, starting first from the data drift detection module. This module, in particular, could help to significantly improve performance in continuous learning.

ACKNOWLEDGMENTS

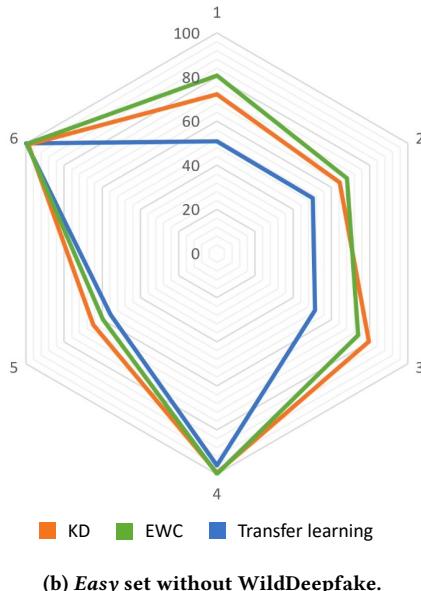
This study has been partially supported by SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU and Sapienza University of Rome project “EV2” (003_009_22).

REFERENCES

- [1] Agil Aghasianli, Dmitry Kangin, and Plamen Angelov. 2023. Interpretable-through-prototypes deepfake detection for diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 467–474.
- [2] Irene Amerini, Aris Anagnostopoulos, Luca Maiano, and Lorenzo Ricciardi Celsi. 2021. Deep learning for multimedia forensics. *Foundations and Trends in Computer Graphics and Vision* 12, 4 (2021), 309 – 457. <https://doi.org/10.1561/0600000096>
- [3] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. 2020. What makes fake images detectable? understanding properties that generalize. In *Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*. Springer, 103–120.
- [4] Riccardo Corvi, Davide Cozzolino, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. 2023. Intriguing properties of synthetic images: from generative adversarial networks to diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 973–982.
- [5] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. 2022. A Continual Learning Survey: Defying Forgetting in Classification Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 7 (2022), 3366 – 3385. <https://doi.org/10.1109/TPAMI.2021.3057446>
- [6] Felipe del Rio, Julio Hurtado, Cristian Buc, Alvaro Soto, and Vincenzo Lomonaco. 2023. Studying Generalization on Memory-Based Methods in Continual Learning. *arXiv preprint arXiv:2306.09890* (2023).
- [7] Jianwei Fei, Zhihua Xia, Benedetta Tondi, and Mauro Barni. 2023. Robust Retraining-free GAN Fingerprinting via Personalized Normalization. In *2023 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 1–6.
- [8] Robert M. French. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences* 3, 4 (1999), 128 – 135. [https://doi.org/10.1016/S1364-6613\(99\)01294-2](https://doi.org/10.1016/S1364-6613(99)01294-2)



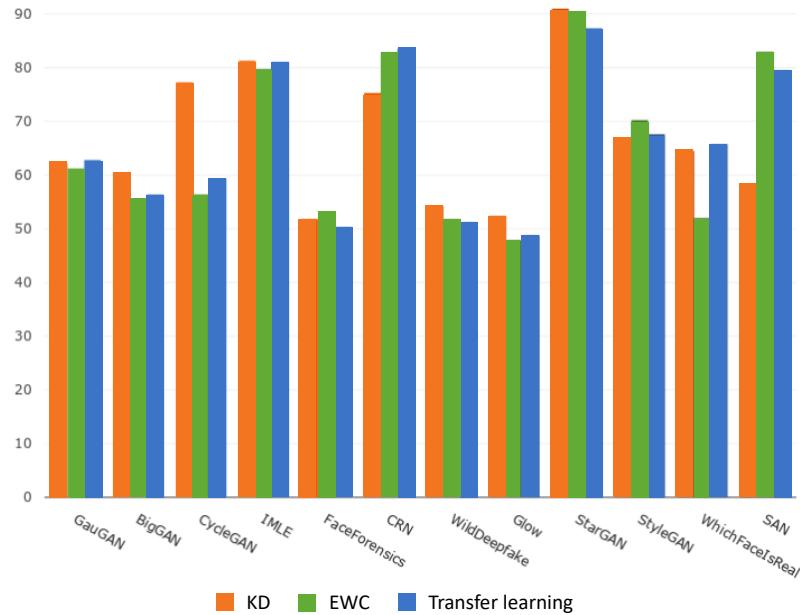
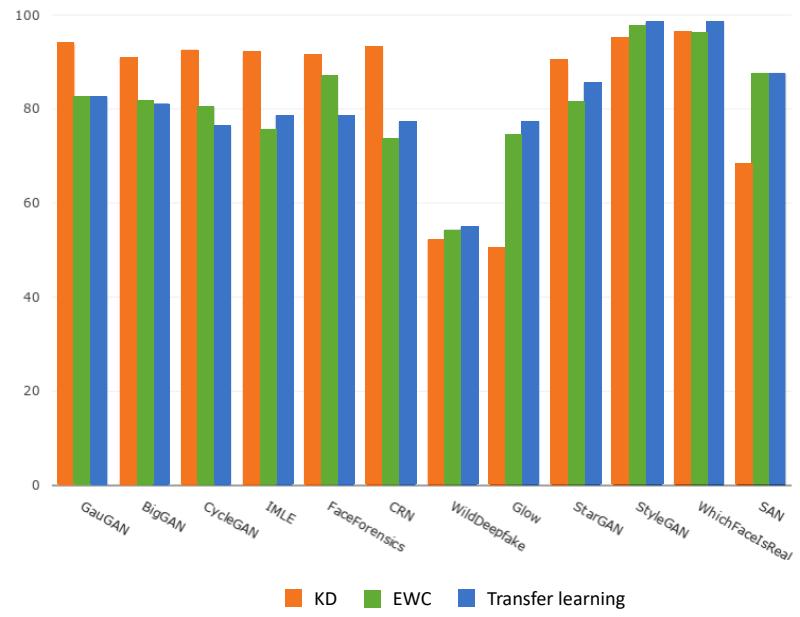
(a) Complete Easy set.



(b) Easy set without WildDeepfake.

Figure 5: The average accuracy of Resnet-50 trained with KD on the full Easy set (Figure 5a) and without WildDeepfake (Figure 5b). Some datasets seem to heavily afflict performance in the continuous learning context.

- [9] Diego Gragnaniello, Davide Cozzolino, Francesco Marra, Giovanni Poggi, and Luisa Verdoliva. 2021. Are GAN generated images easy to detect? A critical analysis of the state-of-the-art. In *2021 IEEE international conference on multimedia and expo (ICME)*. IEEE, 1–6.
- [10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).
- [11] Yan Ju, Shan Jia, Lipeng Ke, Hongfei Xue, Koki Nagano, and Siwei Lyu. 2022. Fusing global and local features for generalized ai-synthesized image detection. In *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 3465–3469.
- [12] Sohal Ahmed Khan and Hang Dai. 2021. Video transformer for deepfake detection with incremental learning. In *Proceedings of the 29th ACM International Conference on Multimedia*. 1821–1828.
- [13] Minha Kim, Shahroz Tariq, and Simon S Woo. 2021. Cored: Generalizing fake media detection with continual representation using distillation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 337–346.
- [14] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America* 114, 13 (2017), 3521 – 3526. <https://doi.org/10.1073/pnas.1611835114>
- [15] Binh M Le and Simon S Woo. 2023. Quality-agnostic deepfake detection with intra-model collaborative learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 22378–22389.
- [16] Chuqiao Li, Zhiwu Huang, Danda Pani Paudel, Yabin Wang, Mohamad Shahbazi, Xiaopeng Hong, and Luc Van Gool. 2023. A Continual Deepfake Detection Benchmark: Dataset, Methods, and Essentials. *Proceedings - 2023 IEEE Winter Conference on Applications of Computer Vision, WACV 2023* (2023), 1339 – 1349. <https://doi.org/10.1109/WACV5668.2023.00139>
- [17] Luca Maiano, Lorenzo Papa, Ketbjano Vocaj, and Irene Amerini. 2022. DepthFake: a depth-based strategy for detecting Deepfake videos. *arXiv preprint arXiv:2208.11074* (2022).
- [18] Francesco Marra, Cristiano Saltori, Giulia Boato, and Luisa Verdoliva. 2019. Incremental learning for the detection and classification of gan-generated images. In *2019 IEEE international workshop on information forensics and security (WIFS)*. IEEE, 1–6.
- [19] Andrei Paleyes, Raoul-Gabriel Urma, and Neil D. Lawrence. 2022. Challenges in Deploying Machine Learning: A Survey of Case Studies. *Comput. Surveys* 55, 6 (2022). <https://doi.org/10.1145/3533378>
- [20] Kun Pan, Yifang Yin, Yao Wei, Feng Lin, Zhongjie Ba, Zhenguang Liu, Zhibo Wang, Lorenzo Cavallaro, and Kui Ren. 2023. DFL: Deepfake Incremental Learning by Exploiting Domain-invariant Forgery Clues. *MM 2023 - Proceedings of the 31st ACM International Conference on Multimedia* (2023), 8035 – 8046. <https://doi.org/10.1145/3581783.3612377>
- [21] Lorenzo Papa, Lorenzo Faiella, Luca Corvitto, Luca Maiano, and Irene Amerini. 2023. On the use of Stable Diffusion for creating realistic faces: from generation to detection. In *2023 11th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, 1–6.
- [22] Muhammad Anas Raza and Khalid Mahmood Malik. 2023. Multimodaltrace: Deepfake Detection Using Audiovisual Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 993–1000.
- [23] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2001–2010.
- [24] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. 2019. FaceForensics++: Learning to detect manipulated facial images. *Proceedings of the IEEE International Conference on Computer Vision 2019-October* (2019), 1 – 11. <https://doi.org/10.1109/ICCV.2019.00009>
- [25] Rudy Semola, Vincenzo Lomonaco, and Davide Bacciu. 2022. Continual-learning-as-a-service (clasas): On-demand efficient adaptation of predictive models. *arXiv preprint arXiv:2206.06957* (2022).
- [26] Luisa Verdoliva. 2020. Media Forensics and DeepFakes: An Overview. *IEEE Journal on Selected Topics in Signal Processing* 14, 5 (2020), 910 – 932. <https://doi.org/10.1109/JSTSP.2020.3002101>
- [27] Jun Wang, Omran Alameyreh, Benedetta Tondi, and Mauro Barni. 2023. Open Set Classification of GAN-based Image Manipulations via a ViT-based Hybrid Architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 953–962.
- [28] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. 2020. CNN-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8695–8704.
- [29] Hanzhou Wu, Gen Liu, Yuwei Yao, and Xinpeng Zhang. 2020. Watermarking neural networks with watermarked images. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 7 (2020), 2591–2601.
- [30] Yibo Zhang, Weiguo Lin, and Junfeng Xu. 2024. Joint Audio-Visual Attention with Contrastive Learning for More General Deepfake Detection. *ACM Transactions on Multimedia Computing, Communications and Applications* 20, 5 (2024), 1–23.
- [31] Yuan Zhao, Bo Liu, Ming Ding, Baoping Liu, Tianqing Zhu, and Xin Yu. 2023. Proactive deepfake defence via identity watermarking. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 4602–4611.
- [32] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang. 2020. WildDeepfake: A Challenging Real-World Dataset for Deepfake Detection. *MM 2020 - Proceedings of the 28th ACM International Conference on Multimedia* (2020), 2382 – 2390. <https://doi.org/10.1145/3394171.3413769>

Figure 6: Average accuracy for Resnet-50 on the *Long* set.Figure 7: Average accuracy for Resnet-50 trained with the multi-task configuration on the *Long* set.