big data and cognitive computing

MDPI

Review

A Survey of Incremental Deep Learning for Defect Detection in Manufacturing

Reenu Mohandas [1,*], Mark Southern [2], Eoin O'Connell [1] and Martin Hayes [1]

[1] Department of Electronic and Computer Engineering, University of Limerick, V94 T9PX Limerick, Ireland
[2] School of Engineering and Enterprise Research Centre, University of Limerick, V94 T9PX Limerick, Ireland
* Correspondence: reenu.mohandas@ul.ie

Abstract: Deep learning based visual cognition has greatly improved the accuracy of defect detection, reducing processing times and increasing product throughput across a variety of manufacturing use cases. There is however a continuing need for rigorous procedures to dynamically update model-based detection methods that use sequential streaming during the training phase. This paper reviews how new process, training or validation information is rigorously incorporated in real time when detection exceptions arise during inspection. In particular, consideration is given to how new tasks, classes or decision pathways are added to existing models or datasets in a controlled fashion. An analysis of studies from the incremental learning literature is presented, where the emphasis is on the mitigation of process complexity challenges such as, catastrophic forgetting. Further, practical implementation issues that are known to affect the complexity of deep learning model architecture, including memory allocation for incoming sequential data or incremental learning accuracy, is considered. The paper highlights case study results and methods that have been used to successfully mitigate such real-time manufacturing challenges.

Keywords: deep learning; incremental learning; continuous learning; catastrophic forgetting; self-healing processes; defect detection; concept drift

Citation: Mohandas, R.; Southern, M.; O'Connell, E.; Hayes, M. A Survey of Incremental Deep Learning for Defect Detection in Manufacturing. *Big Data Cogn. Comput.* **2024**, *8*, 7. https://doi.org/10.3390/bdcc8010007

Academic Editors: Teen-Hang Meen, Charles Tijus, Cheng-Chien Kuo, Kuei-Shu Hsu, Kuo-Kuang Fan and Jih-Fu Tu

Received: 1 December 2023
Revised: 22 December 2023
Accepted: 25 December 2023
Published: 5 January 2024

1. Introduction

Model-based deep learning has long been viewed as the go-to method for the detection of defects, process outliers and other faults by engineers who wish to use artificial intelligence within computer vision-based inspection, security or oversight tasks in manufacturing. The data-hungry nature of such deep learning models that have accompanied the proliferation of AI-enabled data acquisition systems means that the new information that is either captured or inferred in realtime by sensors, IoT devices, surveillance cameras and other high definition images must now be gathered and analysed in an ever smaller time window. Additionally, transfer learning and transformer networks are now providing researchers with the capacity to build pre-trained networks [1], which, when coupled with a deep learning framework, enable the generation of diagnostic outputs that give highly accurate real-time answers to difficult inspection questions, even in those cases where only relatively small datasets are known to exist a priori. The requirement that data need to be independently and identically distributed *(i.i.d)* across the training and test dataset has arisen with the advent of transfer learning and the availability of pre-trained models that can be 'fine-tuned' for a particular task at hand.

More recently, there has been an emerging trend within the literature that proposes the use of generative networks and synthetic datasets for inspection [2]. Such procedures provide a repeatable method whereby a dataset can be augmented in real time with enough process-specific samples so that the model training step can be continuously updated. Numerous data augmentation processes have been proposed for the training phase that optimises the size of the data window required to facilitate accurate decision making in

a resource-efficient manner [3–5]. As an illustration, in [6], the authors claim that models trained using generative adversarial network data and fine-tuned using (10% by volume) randomly selected data gathered in real time from actual MRI analysis exhibits higher performance in tumour segmentation trials.

Classically, deep learning models were trained based on an underlying assumption that all the possible exceptions to be detected were available within the dataset a priori. In such an *offline* learning setup, sufficient numbers of static images would be collected, labelled and classified into fixed sets or categories [7]. The generated datasets would be then further divided into training and test subsets, where the training data would be fed into the network for sufficiently many epochs and test evaluation performed so that a high level of confidence would exist that all possible faults could be reliably detected. Such an approach to defect detection has its roots in the AdaBoost algorithm [8] and papers therein. Limiting factors in such an approach include that the classifier parameters are generally fixed and huge amounts of data are required, making the whole process cumbersome and resource intensive. Furthermore, the response of the model to new error (exception) data is not likely to be robust.

The pre-design phase that is invariably required for offline training is the most important differentiating factor between offline and *online* or *dynamic* approaches to detection or inspection. Online models are tuned using exception data from particular examples of interest rather than simply being restricted to a larger fixed set [9]. Complete re-training of existing models is an expensive task. Although retraining can be achieved in stages using cloud infrastructure while the manufacturing process continues, the streaming nature of data becomes a constraint in retraining the existing network model when new products are added or new process defect information comes to light.

The way in which new classes/tasks are added to a deep learning model once it has been deployed in the field for a specific inspection task is a recurring engineering challenge for the deployment of AI in manufacturing. Figure 1 is an illustration of process cycle without incremental learning. In a practical real-world setting, decision loops based on continuous, temporal streams of data (of which only a narrow subset maybe potentially actionable) must be considered dynamically so that categorisation, exception handling and object identification are not pre-designed or classified a priori. In recent times, the concept of continual or dynamic learning is a recurring significant theme in the literature. Continual learning is the process of learning wherein a new category can be added or processed by the same neural network from a continuous stream of data while at the same time maintaining (or ideally improving) detection accuracy [10]. This concept has been considered by multiple authors, where the overarching principle for process fault detection is one of continual, dynamic learning that has been denoted by authors as *lifelong* or *on the fly*.

A particular focus of this review is a consideration of model-based detection in case studies where a requirement for sustainability through life cycle extension is providing the impetus for the real-time detection, decision making, refurbishment, rework or re-manufacture within a process. The objective is to develop self-healing rather than pushing a 'Bin' or 'End-of-Life' decision within the process. Reliable defect detection is an important step in re-manufacturing, most importantly where inspection fails in (possibly recycled) products that need to be effectively classified so that an improved or refurbished product can be efficiently graded for market or for an appropriate downstream process task. The difference in re-manufacturing as opposed to manufacturing is that a dramatically larger set of variables can affect product sustainability. Moreover, if recycling is a consideration, then fault/defect decisions that are made can be radically different. Defects need to be detected and rework/repair work needs to be predicted, possibly using a deep learning model if appropriate, for every new part or device that is inspected. This requires continuous updates of deployed process models as newer inspection information is gathered. The recycling of devices help manufacturers reduce their carbon footprints in manufacturing [11]. In this domain, there is an increased premium placed on dynamic, continual or incremental

learning that reduces mislabelling, handles new exceptions reliably as they occur and is robust to a high degree of variation in products that are inspected in real time.

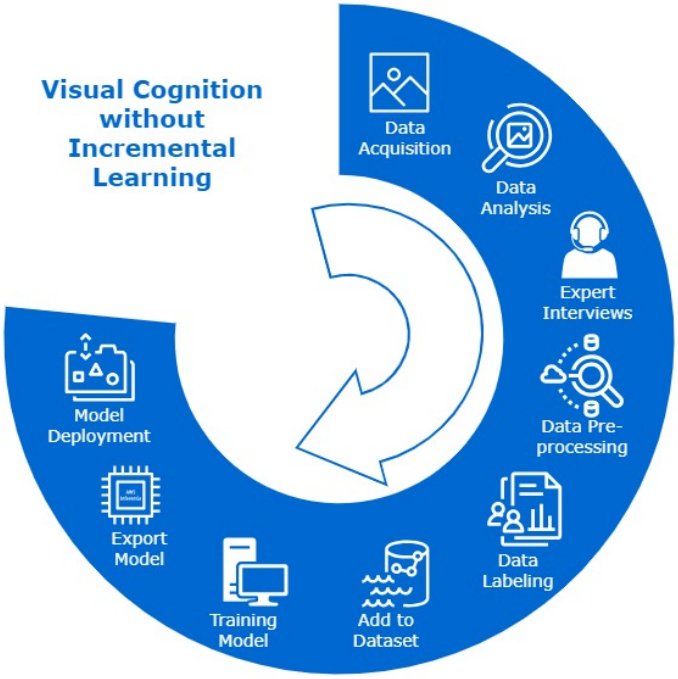

Figure 1. Visual cognition engine development process cycle using deep learning without incremental learning. The process stages end with model deployment, and the lifelong learning process is incomplete without incremental learning algorithms to update the deployed model-based detection system.

This review article is laid out as follows: First, a general summary of inspection algorithms that use incremental learning is presented in Section 2, *Literature Study*. This section discusses incremental learning in manufacturing deployments, and also the question of imbalance or bias in data handling that can occur due to dynamic streaming. Next, in Section 3, a comprehensive analysis of the use of incremental learning in model-based machine learning frameworks that have been deployed for defect detection is presented. This section also discusses process complexities such as *catastrophic forgetting* and ensuing mitigation strategies that have been reported. An analysis of the deployment of incremental learning in applications where processing takes place *at the edge* or the use of edge devices is explicitly considered in Section 4. Section 5 considers process prediction and operator training challenges with a particular emphasis on the drift phenomenon problem that is a pressing concern in incremental learning deployments for process prediction applications. In Section 6, the performance and applicability of different incremental learning algorithms are considered, with a particular emphasis on the size of the datasets that are necessary and the related efficiency of the processing that takes place. Finally, Section 7 presents some conclusions and reflections on future research challenges in this space.

2. Literature Study

Against the backdrop of data engineering challenges that arise in cyber–physical systems and automated inspection systems within an Industry 4.0 setting, there is an increased requirement for intelligent agents and processes that can adapt and update dynamically in uncertain environments [12]. A number of deep learning frameworks have been proposed in the incremental learning literature as detection and/or classification steps within a process that are trained on *i.i.d* datasets, which use batch processing of well-labelled data. The challenge is one of adapting a model in real time so that it is capable of reliably incorporating new information while at the same time being stable in terms

of process performance based on existing information. Table 1 provides a timeline for recent advances within incremental learning that have particular impact in relation to manufacturing. The concept of lifelong learning in the context of visual cognition engine development process cycle is illustrated in Figure 2.

Table 1. Overview of research articles in the area of incremental learning in recent years.

Year	Contribution to Incremental Learning	Research Article Title
2009	Tracking modelling detection—adaptive tracking with online learning	Online Learning of Robust Object Detectors during Unstable Tracking
2015	Sparse auto-encoder-based framework for feature extraction and active learning by gradient descent	A Continuous Learning Framework for Activity Recognition Using Deep Hybrid Feature Models
2016	Elastic weight consolidation	Overcoming Catastrophic Forgetting in Neural Networks
2017	Hedge backpropagation (HBP) method for DNN parameter update	Online Deep Learning: Learning Deep Neural Networks on the Fly
2017	New loss function using distillation loss to minimise catastrophic forgetting	Incremental Learning of Object Detectors without Catastrophic Forgetting
2019	Comparison of sequential learning tasks based on DNNs	A Comprehensive, Application-Oriented Study of Catastrophic Forgetting in DNNs
2019	Knowledge distillation to mitigate catastrophic forgetting	RILOD: Near Real-Time Incremental Learning for Object Detection at the Edge
2020	Modified cross-distillation loss function method to alleviate catastrophic forgetting and concept drift	Incremental Learning In Online Scenario
2020	Method to efficiently store tensor-quantised representation of input images and replaying them	REMIND Your Neural Network to Prevent Catastrophic Forgetting
2021	Online continual object detection benchmark using continuous data streams in dynamic environments	Wanderlust: Online Continual Object Detection in the Real World

Figure 2. Visual cognition engine development process cycle using deep learning-integrated incremental learning completes the cycle of *lifelong learning*. The ideal concept of the lifelong learning cycle involves process stages including the steps for incremental learning algorithms. This sequential learning system should maintain high accuracy of the deployed model-based detection system and update the system with new categories.

In this context, the authors have considered analogous, subtly different incremental learning variations such as lifelong supervised learning, continual learning, open-world learning and online continual learning so that an overarching picture of dynamic, incremental learning can be established. This concept of adding new classes to an existing model from streaming data is schematically represented in Figure 3. The *continual* process of intelligent adaptive learning from dynamic streams of data where new 'teachable moments' arise non-deterministically is of special interest in manufacturing since any new exception data becoming available will need to be acted on and assigned, generally aperiodically. Such assignment is a challenge that resides at the core of all the aforementioned dynamic learning paradigms. In particular, the focus here is on case study examples where new exceptions and categories are learned in real time so that mitigation of the phenomenon that has been identified as 'catastrophic forgetting' [10,13–18] is considered.

Consider an ideal case, where for any Class C , where C_i is the i_{th} instance of that class where learning takes place incrementally:

$$C = C_1, C_2, C_3, \ldots C_i, \ldots C_n \tag{1}$$

Thus, the process of adding information continues incrementally until no new information can be gathered. Catastrophic forgetting refers to the practical loss of information that might occur due to model update or retraining in the attempt of adding a new class to the existing model so that the size of the class window C is constrained. In a situation where information from only the n most recent instances of the class is stored reliably, then any information about the class that has been learned prior to this window might be lost. When a new class, C_n is learned, the performance on $C_n - 1$ drops drastically.

In the context of incremental learning in the manufacturing process, the addition of a new class into a detection system reduces the accuracy in performance for a previously learned class. In case of mobile phone defect detection, consider a detection model trained on surface scratches on the phone screen. In an event of a new class being presented to an already existing high performing model, the process of fine-tuning and updating the previously working model with the new class (cracks) is adopted. This update in the class window of the model-based detection system further leads to rapid reduction in accuracy of detection on phone screen scratches, the previously learned class. This is where the research for incremental learning is significant in a manufacturing use case. A deep learning model deployed in a factory setting will be presented with new defect formats or types, and the model needs to be equipped to incrementally learn the new classes and maintain the high detection rate on all the class categories learned.

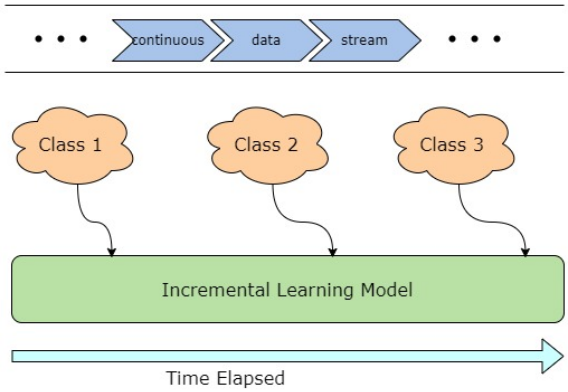

Figure 3. High-level block schematic representing the concept of incremental learning from a continuous stream of data.

Window sizing within continual learning and the necessary size of the training window for a specific task has been considered by many authors [19] and references therein. Strong consensus has emerged that the problem of catastrophic forgetting needs mitigation

measures [20]. Lifelong learning has been proposed by Thrun et al. as early as 1995, where a robot agent is desired to optimise to a different control policy for learning a new function and new environment where functionality can be maximised over time [13]. Classification attributes such as *knowledge bottlenecks* where categories in offline learning need to be pre-designed, *engineering bottlenecks* where sufficient data on each category need to be collected and *new data collection methodologies* such as sensor systems, image acquisition systems and further concerns about complexity and maintaining precision of once-developed robotic systems have all been considered. More recently, authors have considered how decisions that are taken are orchestrated at the edge interface between a physical machine and cloud infrastructure that might be deployed to support the process.

Knowledge distillation and replay-based methods were a feature of early case study experiments in incremental learning. In knowledge distillation, learned parameter values, referred to as 'knowledge' parameters, are transferred from one neural network model to a second neural network model by training the new model on a transfer set, despite (or indeed overcoming) any differences that may exist in model architecture [21]. Maintaining process knowledge when engaging in model order reduction has been considered by many authors. Caruana et al. [22] has referred to the concept of model compression and has introduced knowledge distillation, which is now a proven method for the transfer of learned parameter data among deep learning models. This transfer is made by using class probabilities produced by larger models as soft targets in the newer, smaller model, thereby achieving the generalisation ability in the smaller model otherwise harder to achieve through training [23].

Model compression is a concept inherently different from model order reduction. Model compression refers to the process of effectively reducing the network size in memory leading to faster inference. This reduction in size is achieved by change in model quantisation, adjusting the floating point variables required for inference. Redundant connections in an otherwise over-parameterised model can be pruned to reduce model size, but the number of hidden layers will remain the same. Thus, model compression is different than the concept of model order reduction. In the context of incremental learning, model compression comes into context with the adaptation of knowledge distillation as a method to alleviate catastrophic forgetting.

Replay-based methods are integrated as an effort to alleviate catastrophic forgetting by replaying the previously learned knowledge [24]. One of the essential requirements for replay methods is buffer memory to store new exemplars of new classes/tasks to be learned. The authors in [25] trained a deep neural network with a cross-distilled loss function and approached incremental learning as a four-stage process. The first stage is preparation of training data with representative samples, the second stage is the training process for the selected model, the third stage is fine-tuning with a subset of the data and fourth and final stage is updating the representative memory with samples from a new class. This work is important in the current discussion of incremental learning for the concept of *representative memory*, which performs two significant operations, selection of new samples and removal of unused samples. In a class incremental learning method, termed as IL2M (incremental learning with dual memory) [26], a second memory is introduced to store the statistics of prediction of previously learned classes in an effort to reduce forgetting. Following on from Figure 1, Figure 4 illustrates that by the addition of incremental learning operations, process cycle without incremental learning can be converted into a lifelong learning process cycle.

An end-to-end trainable adaptive expansion network (E2-AEN) network to dynamically generate light-weight modules called adaptively expandable structures (AES) for new tasks, while maintaining accuracy for previously learned tasks has been proposed by Cao and collaborators [27]. The proposed network also includes feature adaptors, which play the essential role of acquiring new concepts and avoid task interference effectively. The network structure is dynamically changeable by the adaptive gate-based pruning strategy that is used to reduce the redundant parameters. This method achieves good accuracy with Pascal VOC [28] and COCO [29] datasets, but the network accuracy has been shown to

be dependent on representation by the backbone network and the large dataset used in pre-training this backbone framework.

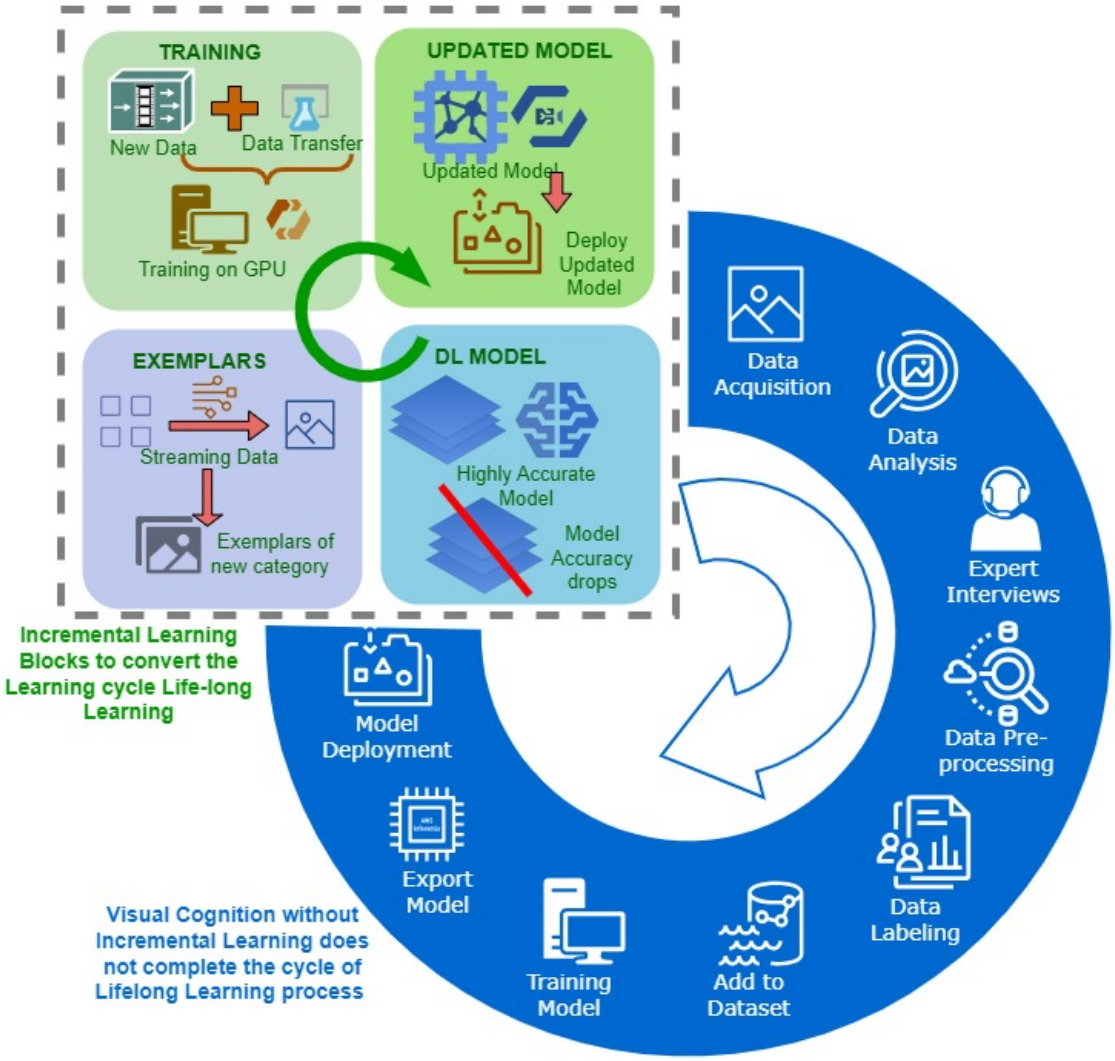

Figure 4. Development of a visual cognition engine using deep learning and the integration of incremental learning blocks to complete the lifelong learning process for a model-based detection system.

Further, end-to-end architecture for class-incremental learning with knowledge distillation in [30] used Faster-RCNN [31] as the backbone and adapted it to incremental learning using domain expansion to include newly added classes of objects and knowledge distillation to maintain accuracy of previously learned classes. The highest accuracy achieved by this method is 72% on newly added classes, but on previously learned classes, the performance is still lower by more than 10% of the original, thereby indicating the challenge of catastrophic forgetting and hence being not feasible for real-time deployment. In [32], incremental end-to-end learning is used for further data collection closer to the data collected by the human individual in an attempt to perform online learning in the autonomous driving domain. The method is reported to achieve 70% accuracy in previously unseen data, but the discussion did not include model architecture or any mitigation technique for loss of previously learned information.

To address the challenge of forgetting, authors in [33] have proposed a unified framework for classification problems where new and old classes are treated uniformly and the average incremental learning accuracy increased by 6% and 13% on CIFAR-100 [34] and ImageNet [35], respectively. The work reduces the imbalance between old and new

classes and proposes a method to preserve effectively previously learned information. The challenge of increasing the type and number of newly learned classes within model-based incremental learning is a significant and central challenge that has been identified by multiple authors. Data imbalance has been identified as an issue when a model is being trained to distinguish between different types and numbers of classes. The authors in [36] addressed this issue using a *bias correction* method, where a diagnosis parameter is used to measure how a classifier is biased towards new data. An optimisation is proposed to measure the bias parameters within a fully labelled and connected layer of the classification model. The method is shown to have achieved high accuracy when tested using the ImageNet [35] and MS-Celeb-1M-10000 [37] datasets, but issues have been shown to exist for examples where smaller class numbers are known to exist. The problem of dataset imbalance requires particular attention when wishing to apply incremental learning in inspection and is discussed in Section 2.1.

2.1. Treating Imbalanced Data in Inspection

When it comes to carrying out inspection, a myriad of non-destructive evaluation (NDE) methods have been reported. These can include, inter alia, material inspection or defect detection using techniques such as ultrasonic inspection, or any computer vision techniques that do not cause damage to the component under inspection [38]. Apart from the increased efficiency in defect detection, deep learning frameworks are not widely used in this domain, owing to the difficulty in augmenting trained models, which require significant retraining and redeployment. The authors in [39] analyse the case of anomaly detection in video surveillance using spatio-temporal auto encoder networks to illustrate the difficulties in detection of previously unknown anomalous behaviors, even to human operators. The complexities in detecting and dynamic adaptations to new defect categories and further evolving behaviors are challenged by the limitations in incremental learning methodologies.

Several studies have found that, when considering traditional classifiers with predefined categories, machine learning algorithms showcase better performance in classification even with imbalanced data, significantly simplifying the process of data cleaning and balancing [40]. Data imbalance in defect detection is a particular issue when the number of defective components is low as a proportion of an overall batch size. In an effort to mitigate the risks due to non-stationary imbalanced data streams, Chen et al. [41] proposed a recursive ensemble approach (REA) where they estimate the similarities between the minority/defective class in previous and current batches. Traditional classifiers were built on the assumption of equal/fair distribution of instances of all different classes identified in the dataset [40]. When this is not true within a sample set, the question of corrective action or model updates becomes more complex.

A variety of data sampling methods have been proposed to handle imbalanced datasets by balancing the sample distributions for the classes that are under consideration [42]. Clearly, such a balancing approach is difficult to achieve successfully in real time, particularly when quite small batch size windows can exist. Under-sampling eliminates random samples of majority classes in an attempt to balance the class instances within the dataset. Drawbacks with this approach are known to be an increased risk of missing important instances or exceptions in the dataset. Random over sampling has been used as a method to increase the occurrences of random minority instances. The engineer can choose to control oversampling rates, particularly when increased or novel failure classes are identified at the output layer. The Synthetic Minority Oversampling Technique, SMOTE [43], is a synthetic sampling data generation technique where the kth-nearest neighbour of every minority sample is calculated, followed by taking random samples according to the over-sampling rate set by the engineer. A new minority class can be generated by interpolating between the minority class instances and selected neighbours, particularly when new defects are identified at the output layer. In synthetic sample data cleaning, a Tomek-Link is proposed as an under-sampling method in Batista et al. [44] where Tomek-Link is denoted as the distance between the two samples under consideration, using the methodology proposed

in [45]. The Tomek- Link method removes instances belonging to majority classes from the task window until all minimally distanced pairs of nearest neighbour points belong to the same class [40]. In [46], an AI-tuned application example that combines SMOTE and Tomek-Link generates better results on imbalanced datasets for an inspection process [46].

Another way of dealing with imbalanced data is referred to in the literature as *cost sensitive learning*, where each misclassification is assigned a cost within a classification matrix. By setting diagonal matrix elements to zero, the effect of a misclassification error can be computed [47]. Boost algorithms have been proposed as another type of classification cost sensitivity method, where weights on misclassifications are increased and decreases on the weights for correctly classified examples are considered in each iteration. This has been shown to improve classification performance especially on rare inspection classes that are more often misclassified due to imbalance (or bias) in the data. Ada-Boost and Rare-Boost are further developments that add weights to misclassified samples, Rare-Boost weighs the proportion by which false positives are distinguished when measured against true-positive and true-negative samples.

To address the imbalance in text data for incremental learning, Jang et al. [48] have proposed a training architecture, sequentially targeting where the entire training data corpus is divided into mutually exclusive partitions to balance data distribution and adapting it to predefined performance distribution targets. A target distribution in this context is a distributional parameter setting where the trained model is tuned to achieve the best learning outcome. In an imbalanced setting, where there is no pre-defined target distribution, all classes will be given equal importance. The need for a pre-defined target distribution for so-called *forced* incremental learning has been observed to yield a model that gives better performance while incrementally learning individual tasks than with a learning process for multiple tasks that are taken together in static fashion [48].

3. Incremental Learning Frameworks for Deep Learning

Deep learning models are often trained using standard backpropagation, where the training corpus that is available is fed in its entirety into a model in batches. This is not a favoured approach in defect detection where new defects can arise aperiodically in a manufacturing setting. The problem is exacerbated if re-manufacturing is required. In areas where data are available as a dynamic stream, the amount of data will incrementally grow, making the storage space insufficient. Network depth is an important factor when dynamically learning complex patterns in an incremental fashion. Deep, parameter-rich models face the problem of slow convergence during the training process [49], but data-driven real-time inspection methods powered by deep learning models have been shown to exhibit significantly improved efficiency in manufacturing use cases [50].

3.1. Incremental Learning in Manufacturing

With the advent of intelligent machines, human-machine collaboration in decision making has evolved into a 'peer-to-peer' interaction of cognition and consensus between a human and the machine, contrary to the 'master–slave' interaction, which existed decades ago [51]. Incremental learning benefits significantly from new-found methods of collaboration based on cognitive intelligence. Human-in-the-loop hybrid intelligence systems codify required human intentions and future decisions for the downstream tasks, which cannot be directly derived by machines. Having built automated production lines and factory settings, there has been the realisation that humans are necessary to provide an effective control layer, yet humans are often considered as an external or unpredictable component in an ML loop [52]. This is where human-in-the-loop oversight becomes important so that operator safety mechanisms can be safely incorporated within inspection loops. When humans interact with machines in heavily automated Industry 4.0-type factory settings, data gathering, storage and distillation are required to transform data into information for modelling purposes, thereby facilitating further automated inspection and the creation of intelligent, self-healing systems [53,54]. The operation of transforming streaming data into

trainable information by selection of exemplars representing new class/category needs expert human oversight. The training stage incorporates storage of processed data, methods to alleviate catastrophic forgetting and further training with new categories. Schematic representation of incremental learning operations for manufacturing environment is given in Figure 5.

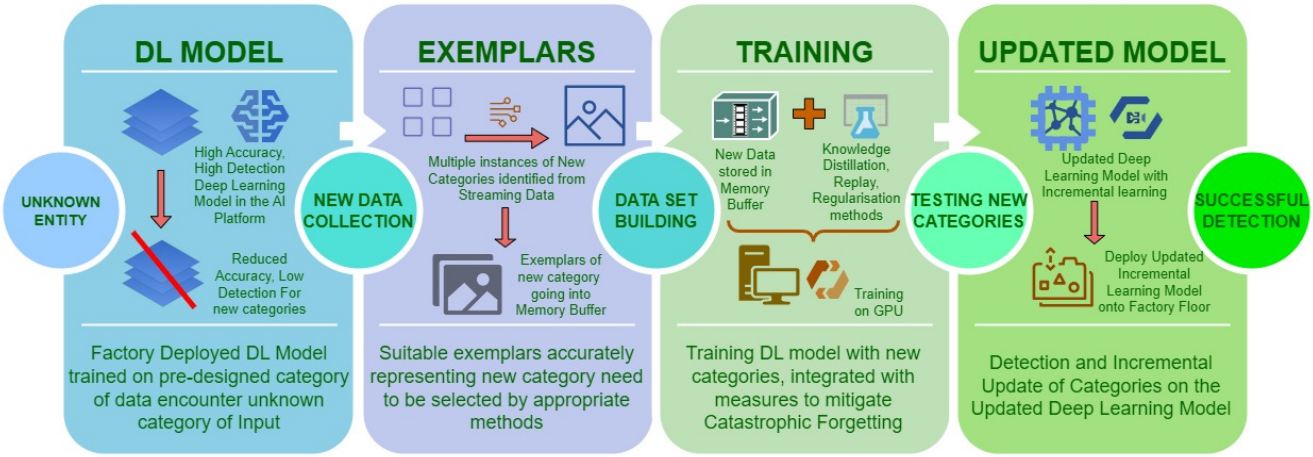

Figure 5. Representation of incremental learning in a visual cognition engine using deep learning.

The challenge of the proportion of defective components being significantly lower than healthy parts during inspection and reliable deep learning model update with the new types of products and defects in the manufacturing setting has been considered in [55]. Current trends appear to focus on the concepts of model updating, retraining and online learning that are capable of updating deployed models online in real time, even when new recycled products are added into a production line. The one-stage detector framework EfficientDet, presented in [56] proposes a family of eight models (D0–D7) that can be used used in a particular deployment experiment depending on the availability of resources, accuracy and complexity of the data. Minimum amounts of quality data are needed to update a particular model in real time if stipulated model accuracy is to be maintained for a manufacturing process. [57] observes that the frequency of data generation and storage in a smart manufacturing setting needs to be determined a priori in any modelling experiment. Long term data averaged at appropriate intervals are best suited for modelling rather than a high volume of rapid data points, which demand more storage capacity.

Intrusion detection and network traffic management systems is another sector that uses online learning for real-time identification of newer threats based on traffic flow patterns [58]. This idea of online learning is also crucial in a factory floor with automated systems, sensors and IoT devices. In [59], a restricted Boltzmann machine (RBM) and a deep belief network are used in an attempt to learn and detect new attacks online. The RBM network is proposed as an unsupervised feature extractor. Restricted Boltzmann machine networks and auto-encoder networks are known to be suitable for feature extraction problems with unlabelled data. The ability of deep auto-encoder networks to learn hierarchical features from unlabelled data is taken advantage of in incremental learning [18]. Shi et al. [60] developed a multi-stage incremental learning approach based on knowledge distillation for online process monitoring. In a machine learning-based process monitoring application, in-situ monitoring and dynamic decision making are shown to be capable only via incremental learning of new anomalies or intrusion attacks.

In [61], authors Lopez-Pez et al. developed a model referred to as GEM, gradient episodic memory, for continual learning, which alleviates forgetting and helps beneficial knowledge transfer to previously learned tasks. The authors were interested in a more 'human-like' learning setting where the number of tasks is large, the number of learning examples is small, the examples are presented to the observer as few times as once, and the performance is measured in terms of knowledge transfer and forgetting. They introduced

two attributes to measure the above criteria, backward transfer (BWT) and forward transfer (FWT). Backward transfer is the effect new learning has on the previously learned tasks, hence, if this backward transfer is negative, it implies loss of data learned and points to catastrophic forgetting. Forward transfer, on the contrary, is the influence of previously learned tasks on the new learning. Positive forward transfer refers to a gain where the previously learned data are helpful in improving the learning process or accuracy of newly learned data. From the experiments, authors inference that the GEM model minimises backward transfer: reduces catastrophic forgetting, but the forward transfer is negligible. A-GEM [62], averaged gradient episodic memory, ensures that the average episodic memory loss over the previous task does not increase at each training step.

Active learning has been in ceaseless research in many classification systems, information retrieval and streaming data applications, in which it has been found to require more storage for the continuous update of these classifiers [18]. Continuous learning problems in machine learning using ensemble models were a common stream of research where new weak classifiers were trained and added to the existing ones as new data are available and outputs from these are weighted to arrive at the final classification decision. The challenges here will be the increase in the number of weak classifiers after more iterations or updates in the model.

Research in neural networks advances by drawing more interest into the concepts of incremental learning and catastrophic forgetting [63]. Categories of incremental learning algorithms developed in recent research are given in Figure 6. Among the two modes of training using gradient descend, online mode could be more appropriately used for incremental training with weight changes being computed for each instance as new training samples are added, as opposed to batch mode training where weight changes are computed over all accumulated instances. The elastic weight consolidation (EWC) algorithm proposed by the authors constrains the weight update for important tasks from previous learning, with the importance of the task being assessed from the probability distribution of data and prior probability being used to calculate conditional probability. This log probability calculated is taken as the negative of the loss function, which then is accounted as the posterior probability distribution for the entire dataset. The posterior probability of previously learned tasks hence extracted form the basis of constraining weight update for these previously learned tasks. The authors in [64] follow the idea of learning without forgetting (LWF) to avoid catastrophic forgetting, which has given promising results in the problem of image classification. In this, the probability vector of output of old classes from the new models is maintained in the same range as that of the old model. Distillation loss on the old classes and cross-entropy loss on the new class are jointly optimised, which in turn gives good performance on the classification task of the new as well as old classes.

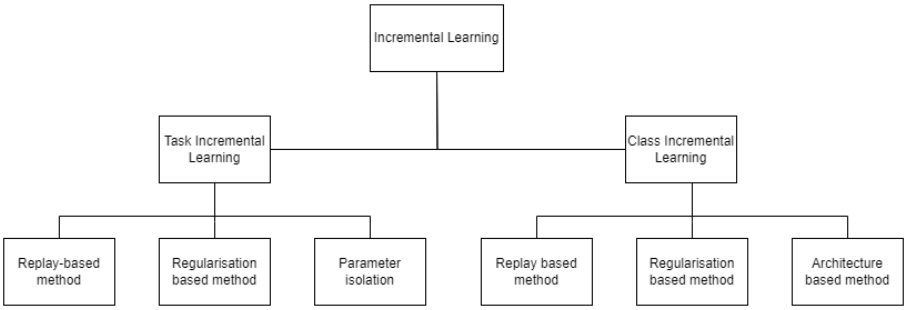

Figure 6. Categories in incremental learning [14,65].

Continual learning methodologies have been classified into three groups in [16]. They are expansion-based, regularisation-based and rehearsal-based methods. Expansion-based methods keep sub-networks or branches or expand existing networks where distinct parameters are given to distinct tasks to identify them during inference. Regularisation-based methods limit the change of parameters for previously learned tasks. Rehearsal-based

methods are mainly focused to alleviate catastrophic forgetting by storing past-task data in a limited buffer and replaying them.

3.2. Catastrophic Forgetting

Catastrophic forgetting has been a well-known concern in the field of deep learning. It is a condition that is known to arise when a neural network loses information that has been previously learned when it is subjected to re-training or a training phase is revisited due to new information gained on subsequent or downstream tasks [24,63,66,67]. Shmelkov et al. describe catastrophic forgetting as "an abrupt degradation of performance on the original set of classes when the training objective is adapted to new classes" [68]. The reason for this degradation in performance of previously learned information is often attributed to the *stability–plasticity dilemma*. McCloskey and Cohen have observed this problem of new learning interfering with the existing trained knowledge in neural networks in their research involving arithmetic fact learning using neural networks in their 1989 study [69]. The researchers conducted experiments based on retroactive interference observed in human learners when a neural network is subjected to arithmetic facts. The neural network used in their study was a 3-layer network with 1024 output arithmetic units. Their explanation for this issue was that the new learning builds new propositional structures in the network, which in turn causes disruption in the retrieval of previously learned data. The most important conclusion from these experiments that is still observable in the neural network applications literature is that gradient descent algorithm performance is not ideally suited to the adjustments that are necessary when new actionable data are collected.

3.3. Stability–Plasticity Dilemma in Neural Networks

The stability–plasticity dilemma in neural networks has been studied as another direct reason for a drop in performance of previously learned tasks; stability refers to retaining the encoded previously learned knowledge in neural networks, and plasticity refers to the ability to integrate new knowledge into these neural networks [14]. Effects of incremental learning on large models and pre-trained ones have been studied by Dyer et al., and their finding was that compared to the randomly initialised models, large pre-trained models and transformer models are more resistant to the problem of catastrophic forgetting [70]. Improved performance in deep learning models has been undeniably tied to the larger size of the training dataset and deeper model size. The empirical studies have been largely conducted on language models, GPT [71] and BERT [72], which are pretrained using the large corpora of natural language text and thereby falls into category of unsupervised pretraining. As opposed to language models, image models are pre-trained in a supervised manner.

For the question of task-incremental or task-specific learning that often occurs in AI-enabled inspection, a neural network will receive sequential batches of data assigned to a specific task, the second is class-specific learning, for which the neural networks are most widely used equally in research and industry. The authors further divide task-specific incremental learning into three categories based on the storage and data usage in sequential learning: (a) replay method, (b) regularisation-based method, (c) parameter isolation method. Replay methods store samples in raw format or use generative models to generate samples. The rehearsal method is one among the replay methods where the neural network is retrained on a subset of selected samples for new tasks, but this method can lead to over-fitting. The authors in [14] also check constraint optimisation as another method where constraints are only given for new tasks. Regularisation-based methods are the second type used in incremental learning. The data-focused ones use knowledge distillation to mitigate forgetting where previous task outputs are used as soft labels. In the third type, the parameter isolation method is where more sub-modules or branches could be added to the architecture such that a task is allocated to a specific part, and this part can be masked in further training sessions. In replay-based methods, GEM [61] is a research work that is based on task-incremental learning where new task updates are constrained

so as not to interfere with previously learned tasks. GEM uses a first-order Taylor series approximation to project gradient direction of the previous tasks to probable outlier regions. A-GEM [62] is an improved version of GEM where the average gradient episodic memory loss of previously learned knowledge is reduced. Incremental learning methods have been categorised based on techniques used in Figure 7. The algorithms given as abbreviations in the following Figure 7 are: iCaRL [73], ER [74], SER [75], TEM [76], CoPE [77], DGR [78], PR [79], CCLUGM [80], LGM [81], GEM [61], A-GEM [62], GSS [82], EWC [63], IMM [83], SI [84], R-EWC [85], MAS [86], Riemannian Walk [87], LWF [64], LFL [88], EBLL [89], DMC [90], PackNet [91], PathNet [92], Piggyback [93], HAT [66], PNN [19], ExpertGate [94], RCL [95], DAN [96].

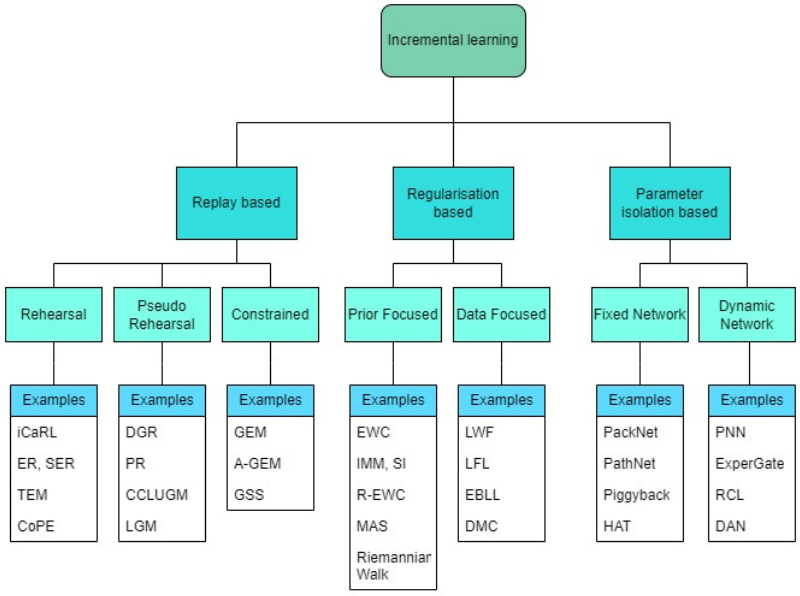

Figure 7. Examples of incremental learning algorithms classified by authors in [14].

3.4. Methods to Alleviate Catastrophic Forgetting

Different methods to address the problem of catastrophic forgetting has been studied. Studies based on architectural change such as adding or reducing branches or dual architectures were also studied alongside replay methods, reducing overlap of the tasks information have all been explored.

3.4.1. Replay Methods

Generative models are used to generate synthetic data or stored sample data are used to rehearse the model by providing this as input during the training of a new task. Due to the reuse of stored examples of data or generated synthetic data, this learning process is prone to over-fitting, hence constrained optimisation has been proposed [14]. iCaRL [73] is one of the earliest replay-based methods developed in 2017, which uses a nearest class prototype classifier algorithm to mitigate forgetting and has been viewed as a benchmark since its introduction. iCaRL used the nearest mean of exemplars for feature representation, reducing the required number of exemplars per class for replay and feature representation from stored exemplars combined with distillation to alleviate catastrophic forgetting. REMIND [65] is another replay-based method where quantised mid-level CNN features were stored and used for replay. Encoder-based lifelong learning (EBLL) [89] is an autoencoder-based method to preserve previously learned features related to the old task as an effort to alleviate forgetting. Replay-based methods such as REMIND [65], ER [74], SER [75], TEM [76] and CoPE [77] are all further replay-based methods. Among replay-based methods is the concern for storage of previous samples and the number of instances used in replay to cover the input space. Generative adversarial networks (GAN)

have been used in generating samples, which could be used in conjunction with real images, and this can be used for replay, termed as *pseudo-rehearsal*, which has also been used in incremental learning applications [78–81].

3.4.2. Regularisation Methods

Among the research in regularisation-based methods, De Lange et al. [14] divided it into two types: data-focused methods and prior-focused methods. Parameter regularisation and activation regularisation methods are the ones already researched to mitigate the problem of loss of previously learned information [64,85,87,88,90]. Data-focused methods use knowledge distillation from a previous model into the new model trained for the new task. The knowledge distillation process is used here to transfer knowledge, and it also helps mitigate the catastrophic forgetting during training for new tasks. The output from the model trained for a previous task is used as a soft label for those previously learned tasks. Regularisation-based methods vary the plasticity of weight depending on the importance of the new task as compared to the old task, previously learned [65]. Methods such as EWC [63], GEM [61] and A-GEM [62] use regularisation methods, but in GEM and A-GEM, regularisation along with replay methods are used. In EWC, weights are consolidated into a Fisher information matrix, which gives parameter uncertainty based on tractable approximation. Compared to EWC, synaptic intelligence (SI) [84] collects individual synapses-based task-relevant information to produce a high-dimensional dynamic system of parameter space over the entire learning trajectory in the effort to alleviate catastrophic forgetting. MAS [86] is a regularisation-based method where changes to previously learned weights are regulated by penalising parameter changes using hyper-parameter optimisation, where importance of weights are estimated from the gradients of the squared L2-norm for output from previous tasks. IMM [83] matches moments of posterior distribution of neural networks in an incremental priority for first and second training tasks and uses further regularisation methods including weight transfer, L2-norm-based optimisation and also dropout methods [63].

3.4.3. Parameter Isolation Methods

Parameter isolation methods isolate and dedicate parameters for each specific task to mitigate forgetting. When new tasks are learned, previous tasks are masked either at unit level or at parameter level. This strategy of using a specific set of parameters for specific tasks may often lead to restricted learning capacity for new tasks [14]. New branches can be grown for new tasks given there is reduced constraint in network architecture. In such cases, parameter isolation is achieved by freezing previous task parameters [92–96]. PackNet [91] assigns network capacity to each task explicitly by using binary masks. In hard attention to task [66], task embedding is implemented by the addition of a fraction of previously learned weights to the network, which then computes conditioning mask using these high-attention vectors. The attention mask is used as a task identifier for each layer and utilises this information to prevent forgetting. In PNN [19], the weights are arranged in column-wise order respective to the new task with random initialisation of weights. Transfer between columns is enabled by adaptors, which are non-linear lateral connections between new columns created with each new task, thereby reducing catastrophic forgetting.

In a study related to class incremental learning, the authors of [97], Xu et al., divided the methods into three categories: (a) replay-based method, (b) regularisation-based method and (c) architecture-based method. The methods are tested on a deep learning-based computer vision for a hyper spectral image (HSI) use case example. The first two categories have been previously considered as task-incremental learning approaches. The third category is denoted as an architecture-based method. This method is a class of parameter-isolation where new branches can be created and a new learning phase is created to distinguish new and old inspection tasks. Progressive neural networks [19], dynamically expandable networks [98], adaptation by distillation [99] and dynamic generative memory [100] are a few methods using expanding architecture-based methods to miti-

gate catastrophic forgetting. The sub-network for each classification task is learned to prevent catastrophic forgetting. Knowledge distillation is the most important addition to the incremental learning framework brought forward by the authors where classification performance of old classes is maintained by training the network with old knowledge. The total loss is obtained from so-called soft targets that have been developed on old classes, and network features of old classes are saved to avoid performance degradation in these old classes when new classes are created.

4. Incremental Learning for Edge Devices

In 2019, Li et al. studied incremental learning for object detection at the Edge [101]. The increased use of deep learning models for object detection on Edge computing devices was accelerated by one-stage detectors such as YOLO [102], SSD [103] and RetinaNet [104]. A deep learning model deployed at the Edge needs incremental learning to maintain the accuracy and robust performance in object detection in personalised applications. The algorithm termed as RILOD [101] is a method for incremental learning where the one stage detection network is trained end-to-end using a comparatively smaller number of images of the new class within the time span of a few minutes. The RILOD algorithm uses knowledge distillation, which has been used by several researchers to avoid catastrophic forgetting. Three types of knowledge from old models were distilled to mimic the output of previously learned classes on tasks such as object classification, bounding box regression and feature extraction. A pipeline for real-time data collection for dataset construction and automatic labelling of these collected images based on category and bounding box annotations have also been developed.

In DeeSIL [105], fixed representations for class are extracted from a deep model and then used to learn shallow classifiers incrementally, which makes it an incremental learning adaptation for transfer learning. Since feature extractors replace real images, memory constrain for new data is addressed, hence making it a possible candidate for Edge devices. Train++ [106] is an incremental learning binary classifier for Edge devices, though it is based on training ML models on micro-controller units. In [107], a task adaptive incremental learning, *TeAM* for convolutional neural networks (CNN) is proposed as a method to transform large CNN models into optimised forms as to work in Edge devices and a global aggregation of collaborative models on local devices into a global model, thereby making incremental learning possible. Hussain et al. [108] proposed learning with sharing (LwS) as a method for incremental learning in deep learning framework optimised for Edge devices, which involves cloning of the initial DNN framework except the output layer and freezing all those layers excluding fully connected (FC) layers. These cloned layers and FC layers combined with new output layers are used in the next stage of training, effectively transferring previously learned data. The authors report 75.5% accuracy with Mobile-NetV3 [109] on the ImageNet dataset.

Rapid development of Edge Intelligence with optimisation of deep neural networks led to increased use of model-based detection in computer vision applications. Due to its huge reduction in size as well as reduction in computational costs, the model quantisation is the most widely used optimisation method among all the other types of optimisation and compression techniques for deep learning.

To analyse comparative performance of visual cognition-based deep learning models on a GPU-accelerated device versus a resource-constrained Edge device, Raspberry Pi, we conducted a case study [110]. The emphasis of the study is to assess the detection time and accuracy of deep learning models optimised for Edge functionality. The SSD inception network trained on the INRIA [111] dataset is the reference experiment for the model optimisation criteria. This is included as the first model in both Tables 2 and 3. In 2014, Szegedy et al. and a team from Google proposed GoogLeNet (22 layers), which consists of inception modules. This later came to be widely known as Inception Net [112]. The architecture of these networks were further modified in 2015, which led to versions

Inception-v2 and Inception-v3 [113]. MobileNets were lighter networks and were also designed by Google engineers for mobile vision applications [114].

Table 2. Comparison of SSD models on the GPU vs. the Raspberry Pi used in the person detection case study, evaluated on the basis of detection time and IOU value [110].

Object Detection Model	Time for Detection (ms)	IOU Value
SSD-Inception-v2-coco(GPU)	3.247	0.8034
SSD-Mobilenet-v1-coco(GPU)	2.119	0.7699
SSD-Mobilenet-v2-coco(GPU)	1.811	0.713
SSDLite-Mobilenet-v2-coco(GPU)	1.332	0.6337
SSD-Inception-v2-coco(Rasp-Pi)	2.309	0.7081
SSD-Mobilenet-v1-coco(Rasp-Pi)	1.027	0.7829
SSD-Mobilenet-v2-coco(Rasp-Pi)	0.813	0.7378
SSDLite-Mobilenet-v2-coco(Rasp-Pi)	0.648	0.591

Table 3. Comparison of SSD models on the GPU vs. the Raspberry Pi used in the person detection case study based on precision and recall value of detection [110].

Object Detection Model	Precision	Recall
SSD-Inception-v2-coco(GPU)	1.0	0.8
SSD-Mobilenet-v1-coco(GPU)	1.0	1.0
SSD-Mobilenet-v2-coco(GPU)	1.0	0.8
SSDLite-Mobilenet-v2-coco(GPU)	0.8	0.8
SSD-Inception-v2-coco(Rasp-Pi)	1.0	1.0
SSD-Mobilenet-v1-coco(Rasp-Pi)	1.0	1.0
SSD-Mobilenet-v2-coco(Rasp-Pi)	1.0	0.8
SSDLite-Mobilenet-v2-coco(Rasp-Pi)	0.66	0.8

The detection models were trained on TensorFlow Object Detection API on a local GPU accelerated device running TensorFlow-GPU version 1.14, Python 3.6 and OpenCV 3.4 as the package for image analysis. The GPU used for training was GPU GeForce RTX 2080 Ti, after which TensorFlow-lite graph was exported as the frozen inference graph. This was then converted into flat-buffer format before integrating into Raspberry Pi 4, the resource constrained device used for detection experiments. The Raspberry Pi 4 runs Raspbian Buster-10 OS, with an integrated Raspberry Pi Camera Module V2 for real-time image capture in detection experiment. The Raspberry Pi camera is 8 megapixels, single channel, and has a maximum frame rate capture of 30 fps. The camera module connects to Raspberry Pi 4 via 15 cm ribbon cable which attaches the Pi Camera Serial Interface port (CSI) to the module slots on the Pi.

The inference time of TensorFlow Lite model were compared against that of larger models in the Figure 8, and the results were promising. In industrial application where accuracy is a concern, a 10% reduction in accuracy could lead to higher number of erroneous products, causing huge loss in a mass production system. For an acceptable floor rate of above 70% accuracy within 3 ms detection time, the winning candidate models are the quantised versions of SSD-Inception-v2 and SSD-Mobilenet. SSD-Mobilenetv1, the framework designed for mobile and Edge devices, is the best performing with a detection rate of 78% and detection time of 1 ms in this detection experiment. The IOU values fall to the value of 59% with low precision of 0.66 for SSDLite-Mobilenet model, which makes it unsuitable for industrial application with the previously defined criteria. In a person identification operation, floor rate of detection above 70% is a sufficient and standard engineering performance requirement for cell-based manufacturing environment.

The comparative study was performed to establish that performances comparable to the GPU-accelerated device RTX 2080Ti could be achieved on a Raspberry Pi, a resource-constrained Edge computing device used in the experiment. The Raspberry Pi was chosen as the resource-constrained experimental device because of its versatility to be retrofitted

into any manufacturing setting for dynamic decision making. The factory floor prototype setting in the research facility where the study was conducted further utilised Raspberry Pi in numerous edge processing applications. The developed system was a component of a larger detection system, one of which was an operator safety detection mechanism and hence not stand-alone. The detection algorithm when needing to be retrofitted into larger mechanism needed seamless communication with the counterparts and hence TensorFlow-based operations were preferred to PyTorch.

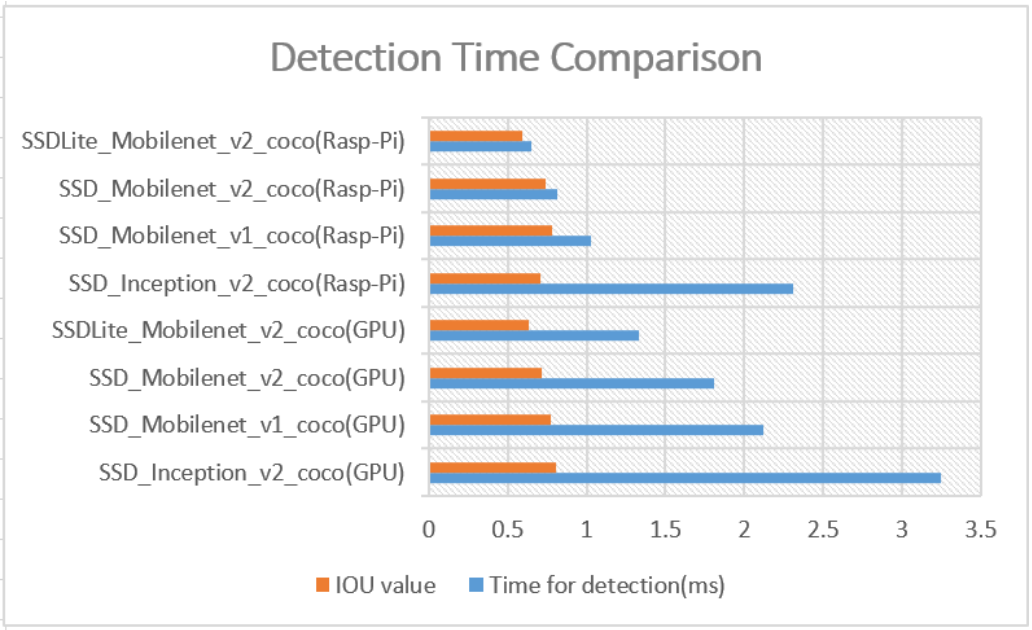

Figure 8. Graph for comparison of detection time vs. IOU value for all the models used. The graph clearly shows comparable detection time in a GPU-accelerated device vs. Edge device (in ms) [110].

5. Process Prediction and Operator Training

Concept drift is another type of problem that occurs in continual learning [15,18], which needs mention for the completeness of challenges in continual learning. Concept drift, also known as model drift, is the change in the statistical properties of a target variable that occurs due to the change in streaming data with the addition of new products/defects [115]. Data drift is found to be one of the factors leading to concept drift. Data drift is the change in distribution of input data instances, which result in variation of predictive results from the trained model. Concept drift can happen over time when the definition of an activity class previously learned might change in the future data streams when the newer models are trained from the streaming data. In the manufacturing setting, maintaining and improving machining efficiency is directly related to the quality of manufacturing end products [116]. Yu et al. studied process prediction in the aspect of milling stability and the effect of damping caused by tool wear in the manufacturing setting. This is an application of incremental learning from the sequential stream of data available and heavily based on concept drift, one of the imminent challenges in incremental learning. The concept drift in this application is the stability domain change, which in turn makes change to the stability boundary. Taking into consideration the time frame in which the sequential data are available, the concept drift is identified as four types:

- sudden drift: introduction of a new concept in a short time
- gradual drift: introduction of new concept gradually over a period of time to replace the old one
- incremental drift: the change of an old concept to a new concept, incrementally over a period of time
- reoccurring drift: an old concept reoccurs after a certain period of time.

In [117], Li et al. made the attempt to combine the incremental learning paradigm with incremental SVM to propose a double incremental learning algorithm for time series prediction. Incremental learning to update predictive models has been studied by the authors of [118] in relation to COVID-19 data to predict variables to characterise the pandemic. This methodology also considers three important points in incremental learning, the complete dataset is not available at the time of creating the model, alleviate catastrophic forgetting and maintaining a balance for the stability–plasticity trade-off. It is also interesting to note that this incremental learning methodology is time series analysis-based as opposed to the deep learning-convolutional neural network-based ones that have been gaining popularity in recent years of research.

Research directly related to incremental learning using deep learning has been conducted by Pierre et al. in [119], where they use correction-based incremental learning in the domain of autonomous vehicles. The algorithm has been tested in relation to truck platooning in simulation and laboratory. The back drop of this project is the inaccuracies that arise from limited/scarce training data near decision boundaries. Driving scenarios such as sudden emergency stop, swerving through multiple curves and drifting off the road are such scenarios under consideration, which can be considered as new instances of interest for the deep learning model. This can be considered as an anomaly in a normal driving situation and could be mapped to a new type of defect occurring in a manufactured component, and the model has to look out for further instances in the manufacturing process.

Correction-based incremental learning augments negative samples into the training set, which were previously classified as positive samples (false positives) to improve the decision boundary. This experiment is also research in incremental learning but without convolutional neural networks for object detection applications. Fully connected layers in neural network architecture are trained in stochastic gradient descent manner with fewer samples that strategically improve the decision boundary for the required task. In the study conducted by Ramos et al. [120], incremental learning based on artificial neural networks are again used to predict industrial electricity consumption by a facility using real-time data and forecasting algorithms. Sequential training data are updated every midnight during the forecast process, where the forecast process is supported by periods split by 5 min intervals. Yu et al. [121] worked with fault diagnosis in the industrial process using incremental learning, again using deep learning framework they termed as the broad convolutional neural network, BCNN. In this method, the abnormal samples collected are combined as a matrix from which non-linear structure and fault tendency are captured by performing convolution operation on the obtained data matrix. Weights of models are calculated from these extracted features to develop the BCNN framework. This methodology permits the feature extraction of any new faults arising in the manufacturing setting and effectively incremental learning on these new faults without retraining.

When considering the paradigm shift from Industry 4.0 to Industry 5.0 use case studies, the impact on people and organisation as well as the technological advances that are proposed must be considered. The main implementation challenges that have been reported in Industry 4.0 applications have been in relation to security, resilience, the ability to withstand disruptions and catastrophic events, operator training and efficient use of digital data from sensors. Industry 4.0 and 5.0 are both aimed at an important dimension of efficient use of energy and technology [122]. Humans create and manage the production systems, hence humans are the main drivers of activities and creators of infrastructure, but the processes in the production will be automated, and any human operator will only be considered as the human-in-the-loop to assist the automated systems. This role of the operator will include selecting the samples from the sequential data acquired by the sensors and labelling or pre-processing for incremental learning techniques. Industry 5.0 prioritises human–machine interaction as opposed to the introduction of robots and automated systems into the manufacturing process in Industry 4.0 [123].

6. Discussion and Analysis

Incremental learning has been researched under two criteria, class incremental learning and task incremental learning. A timeline of state-of-the-art incremental learning algorithms is given in Figure 9. In [24], Mittal et al. studied class incremental learning in an attempt to alleviate catastrophic forgetting. They also used metrics such as 'class-incremental learned models, Class-IL' as referred by them to evaluate the performance. The metrics used are average incremental accuracy, forgetting rate and feature retention. The three important attributes of the class-IL system in their experiment are a memory buffer for storage of examples from old classes, forgetting constraint to mitigate catastrophic forgetting and, the most important element, learning of new classes while balancing old classes.

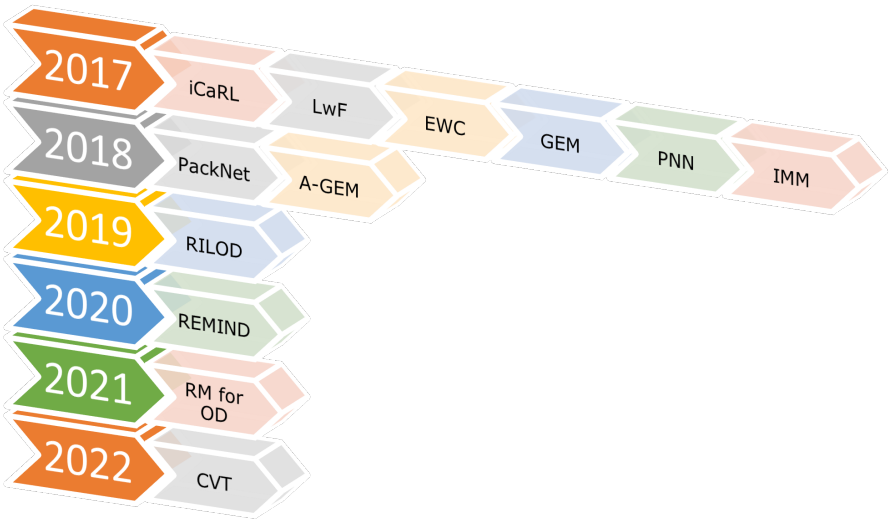

Figure 9. Timeline analysis of state-of-the-art incremental learning experiments.

From the experiments conducted by the authors in [70], catastrophic forgetting has an inverse relation to model scale. Pre-trained vision transformer (ViT) [124] models and pre-trained ResNet both suffer less when the model size is large. Large models suffer less forgetting. The technique used in this analysis, termed as 'forgetting frontier', is a measure of the maximum performance on new data learned for a given stable model performance for old data. A comparison of accuracy loss against model parameter size is given in Table 4.

Table 4. Analysis of the effect of model scale on catastrophic forgetting.

Models	Parameters	Dataset	Accuracy Loss (%)
ViT-xs	5.7 M	CIFAR10	6.5
ViT-Base	86.7 M	CIFAR10	<0.5
ResNet18	11 M	CIFAR10	40
ResNet200	62.6 M	CIFAR10	<0.8

However, Sarwar et al. observed incremental learning as computationally expensive [125]. Their approach focused on using network sharing in the unique clone-and-branch technique, where the cloned layers provide a better starting point to the weights as opposed to randomly initialised ones and hence result in faster learning kernels and faster convergence. The evaluation was based on energy–accuracy trade off, taking into consideration the architecture of the deep convolutional neural networks and complexity of gradient computation and weight update stages.

Regarding the state-of-the-art incremental learning algorithms, there was immense advancement in the research resulting in various benchmark algorithms in the year 2017. iCaRL [73] achieved average accuracy of 68.6%, LWF [64] and EWC [63] with 61%, GEM [61]

with 65.4% and IMM [83] and SI [84] were a few of the most significant among them. Since the inception in 2017, iCaRL [73] has become the bench mark against which numerous class incremental learning algorithms have been analysed. The difference of class-based and task-based is trivial when it comes to learning and weight update criteria from the deep learning perspective. Figure 10 shows the performance analysis of important incremental learning algorithms developed in recent years. In the year 2017 alone, four more significant algorithms were proposed, with average accuracy in the range 64% to 73% among class incremental learning applications. A comparison in accuracy of recently developed incremental learning algorithms with regard to models used and dataset of evaluation is given in Table 5. Table 6 provides important details of benchmark datasets used for pre-training and fine-tuning detection models widely used in the incremental learning research.

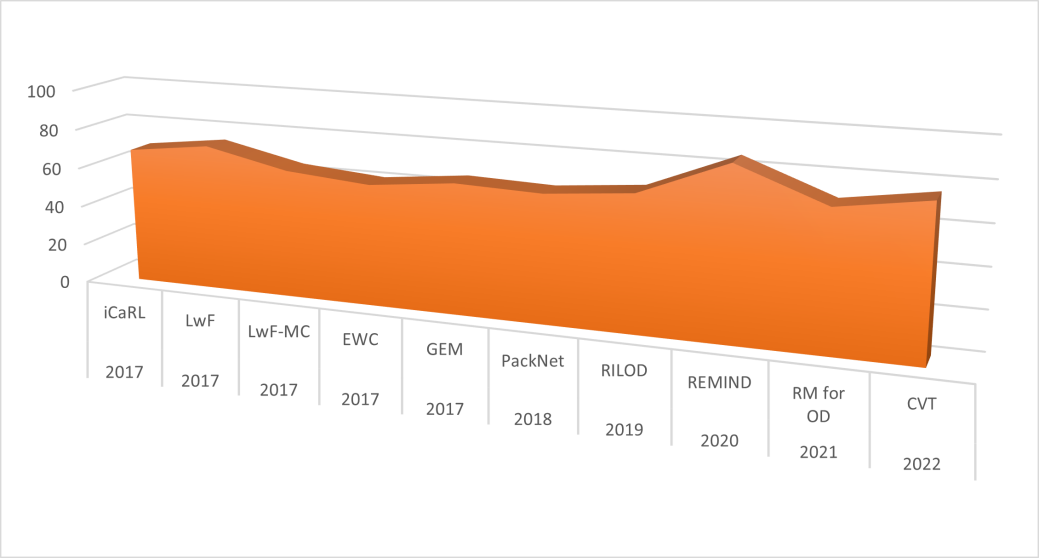

Figure 10. Performance analysis graph for the important algorithms in the domain of incremental learning over recent years.

Incremental learning has also been evaluated based on the loss of information for previously learned tasks. In the correction-based incremental learning approach, the percentage of autonomy and mean time to failure were used in analysis, where autonomy is the percentage of time the system operates without the need for correction or intervention from a human operator [119]. The mean time to failure is the average time the system operates without failure between correction or intervention.

Table 5. Comparison of recent research in incremental learning algorithms based on the dataset used and average accuracy.

Method	Models	Dataset Used	Average Accuracy (%)
A0F0 [1] [70]	Activity learning framework	KTH	98.0
A1F1 [1] [70]		KTH	96.1
A0F0		TRECVID	66.65
A1F1		TRECVID	64.56
A0F0		VIRAT	62.6
A1F1		VIRAT	61.8
A0F0		UCF50	53.8
A1F1		UCF50	44.3
Random Memory [17]	YOLOv5	CORe50	69.08
Tracking Modelling Detection [126]	TMD, online detector	Caviar	72, 64

Table 5. *Cont.*

Method	Models	Dataset Used	Average Accuracy (%)
Online Continual Object Detection [7]	Faster RCNN	OAK dataset	40.92
HAT [66]	AlexNet architecture [127]	Permuted MNIST CIFAR100	98.6 90
REMIND [65]	ResNet-18	ImageNet CORe50	85.5 97.8
Fast R-CNN + EWC [68]	B (16–20) B (11–20)	COCO PASCAL VOC	54.8 67.1
CVT [16]	CVT	CIFAR100 Tiny ImageNet ImageNet100	75.76 36.74 42.61
RILOD [101]	RetinaNet	iKitchen	67.9
Partial Network Sharing [125]	ResNet18 ResNet34 DenseNet121 MobileNet	ImageNet	70.73 86.65 64 62
iCaRL [73]	ResNet32	CIFAR100	68.6
Incremental Learning in Online Scenario [15,101]	ResNet18	Food-101	57.83
GEM [61]	MNIST CIFAR100	FCN (100 ReLU units) ResNet18	89.5 61.2
A-GEM [62]	MNIST CIFAR100	FCN (100 ReLU units) ResNet18	89.1 62.3

[1] A0F0—no active learning and infinite buffer; A1F1—active learning and fixed buffer.

Table 6. Details of dataset used: balanced benchmark datasets and fine-tuning datasets.

Dataset for Pre-Training Backbone Architecture			
Dataset	**Total Images**	**Train Set & Test Set**	**Categories**
ImageNet	14,197,122	80,000 noun synsets in WordNet, 500–100 images in each synset [35]	5247
Pascal VOC	10,000,000	The train/val data have 11,530 images with 27,450 ROI annotated objects and 6929 segmentations [28]	10,000
COCO	328,000	165,482 train, 81,208 val and 81,434 test images [29]	91
Google Open Images V4	9,178,275	30,113,078 image-level labels, 15,440,132 bounding boxes, 374,768 visual relationship triplets [128]	600

Table 6. *Cont.*

Dataset for Pre-Training Backbone Architecture			
Dataset	**Total Images**	**Train Set & Test Set**	**Categories**
JFT	300 M	375M labels, on average each image has 1.26 labels [129]	18,291
Dataset for Fine Tuning and Experimentation			
MNIST	70,000, 28 × 28	60,000—Train images, 10,000—Test images	10
Fashion MNIST	70,000, 28 × 28	60,000—Train images, 10,000—Test images	10
CIFAR-10	60,000, 32 × 32	50,000—Train images, 10,000—Test images	10
CIFAR-100	60,000, 32 × 32	500—Train images, 100 test images per class.	100 (20 superclass)
Tiny ImageNet	100,000, 64 × 64 RGB	500—Train images, 50—val and 50 test images for each class	200
Core50	164,866, 128 × 128	11 sessions × 50 objects × (300) frames per session [130]	50 objects, 10 class
Dataset for Fine Tuning and Experimentation			
SVHN	604,300, 32 × 32 RGB	73,257—Train images, 26,032—Test images and +531,131 train samples	10 (Digit classification)

The discussion on incremental learning cannot be completed without accounting for the benchmark datasets used in pre-training complex, cumbersome deep learning architectures for faster convergence and better generalisation. The deep learning frameworks serve as the backbone for tasks such as detection, segmentation and classification [131–133]. The most widely used method of pre-training as an initialisation method for computer vision tasks is the supervised pre-training [133] using ImageNet with 1.2 M images [35]. Further datasets used in pre-training deep learning architectures include COCO (Microsoft COCO: Common objects in context) [29], PASCAL-VOC [28], OpenImages [128] and JFT [129], which was an internal Google dataset with more than 300 M images that are labeled with 18,291 categories, later published in 2017 [21].

While there is an ongoing debate whether pre-training is necessary for all types of tasks, incremental learning approaches in the literature are found to have used pre-trained models, which were later fine-tuned for specific detection or classification experiments. The datasets used in research studies included in this review article were generic datasets such as MNIST [134], FashionMNIST [135], SVHN [136], CIFAR-10 [137], CIFAR100 [34], Tiny ImageNet [138], which is a strict subset of the large ImageNet dataset, CORe50 [130] and a few task-specific datasets available. MNIST is a hand-written digits dataset. Task agnostic incremental learning is yet in the infancy stage of experimentation, and the incremental learning accuracy on real-world datasets is very low, which implies that real-world experimentation of this technique is under process or not yet available for the research community. The classes/tasks being learned from streaming data can also lead to concept drift as discussed above when there is change in data distribution. Accounting for all those factors, despite all the different types of methods and models studied in the area of incremental learning, there is no clear winner model that can produce state-of-the-art comparable results in an online real-time deployment [15].

7. Conclusions

This paper is a review into the continuous learning aspect of deep learning models and common challenges that arise in inspection and summarises recent research to mitigate such challenges. Continuous learning is a significant research concern for AI-enabled inspection, and model-based detection challenges are becoming ubiquitous in visual cognition aspects in manufacturing. Typically, research efforts have focused on mitigating specific challenges such as catastrophic forgetting, storage and replay of exemplars from previous tasks and the computation of weights that are applied to new classes that can develop over time. More recently, the literature has focused on computational challenges associated with increasingly complex datasets, structures and architectures that arise in modern manufacturing. This leads to uncertainty in relation to knowledge requirements and also incremental learning challenges from the often large amounts of sequential data that are available. This area of research is still in its infancy, and there is no general agreement on the benchmark testing procedure that should be adopted for a particular inspection challenge. Although a wide range of deep learning techniques exist for object detection, there exists no de facto method that can be generally applied to the question of incremental learning in a defect detection process step that incorporates the use of computer vision. Defect detection and analysis poses challenges for incremental learning algorithms, and in the future, equal emphasis will need to be placed on the twin challenges of how a deployed neural network will be updated dynamically and also reducing the impact of catastrophic forgetting in such use cases.

Author Contributions: Conceptualisation: R.M. and M.H.; Writing—original draft preparation, R.M.; Writing—review and editing, R.M., M.S., E.O. and M.H.; Funding acquisition, M.S. and M.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work is completed with support from Predict Project and Confirm, a Science Foundation of Ireland research center in Smart Manufacturing hosted by the University of Limerick. The authors would also like to acknowledge resources made available from Science Foundation Ireland through the grant award (Project No: 16/RC/3918).

Data Availability Statement: The authors confirm that the data supporting the findings of this study are available within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
AI	Artificial Intelligence
DL	Deep Learning
FC	Fully Connected
IL	Incremental Learning

References

1. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A survey on deep transfer learning. In Proceedings of the Artificial Neural Networks and Machine Learning—ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, 4–7 October 2018; Proceedings, Part III 27; pp. 270–279.
2. Gonog, L.; Zhou, Y. A review: Generative adversarial networks. In Proceedings of the 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), Xi'an, China, 19–21 June 2019; pp. 505–510.
3. Khosla, C.; Saini, B.S. Enhancing performance of deep learning models with different data augmentation techniques: A survey. In Proceedings of the 2020 International Conference on Intelligent Engineering and Management (ICIEM), London, UK, 17–19 June 2020; pp. 79–85.
4. Maharana, K.; Mondal, S.; Nemade, B. A review: Data pre-processing and data augmentation techniques. *Glob. Trans. Proc.* **2022**, *3*, 91–99. [CrossRef]
5. Jiang, X.; Ge, Z. Data augmentation classifier for imbalanced fault classification. *IEEE Trans. Autom. Sci. Eng.* **2020**, *18*, 1206–1217. [CrossRef]

6. Shin, H.C.; Tenenholtz, N.A.; Rogers, J.K.; Schwarz, C.G.; Senjem, M.L.; Gunter, J.L.; Andriole, K.P.; Michalski, M. Medical image synthesis for data augmentation and anonymization using generative adversarial networks. In Proceedings of the Simulation and Synthesis in Medical Imaging: Third International Workshop, SASHIMI 2018, Granada, Spain, 16 September 2018; Proceedings 3; pp. 1–11.
7. Wang, J.; Wang, X.; Shang-Guan, Y.; Gupta, A. Wanderlust: Online continual object detection in the real world. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 10829–10838.
8. Qi, Z.; Xu, Y.; Wang, L.; Song, Y. Online multiple instance boosting for object detection. *Neurocomputing* **2011**, *74*, 1769–1775. [CrossRef]
9. Grabner, H.; Bischof, H. On-line boosting and vision. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; Volume 1, pp. 260–267.
10. Chen, Z.; Liu, B. *Lifelong Machine Learning*; Springer: 2018; Volume 1. Available online: https://link.springer.com/book/10.1007/978-3-031-01575-5 (accessed on 29 September 2023).
11. Schimanek, R.; Bilge, P.; Dietrich, F. Inspection in high-mix and high-throughput handling with skeptical and incremental learning. *arXiv* **2023**, arXiv:23284049.v1.
12. Luo, Y.; Yin, L.; Bai, W.; Mao, K. An appraisal of incremental learning methods. *Entropy* **2020**, *22*, 1190. [CrossRef] [PubMed]
13. Thrun, S.; Mitchell, T.M. Lifelong robot learning. *Robot. Auton. Syst.* **1995**, *15*, 25–46. [CrossRef]
14. De Lange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3366–3385.
15. He, J.; Mao, R.; Shao, Z.; Zhu, F. Incremental learning in online scenario. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13926–13935.
16. Wang, Z.; Liu, L.; Kong, Y.; Guo, J.; Tao, D. Online continual learning with contrastive vision transformer. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23 October 2022; pp. 631–650.
17. Nenakhov, I.; Mazhitov, R.; Artemov, K.; Zabihifar, S.; Semochkin, A.; Kolyubin, S. Continuous learning with random memory for object detection in robotic applications. In Proceedings of the 2021 International Conference "Nonlinearity, Information and Robotics (NIR)", Innopolis, Russia, 26–29 August 2021; pp. 1–6.
18. Hasan, M.; Roy-Chowdhury, A.K. A continuous learning framework for activity recognition using deep hybrid feature models. *IEEE Trans. Multimed.* **2015**, *17*, 1909–1922. [CrossRef]
19. Rusu, A.A.; Rabinowitz, N.C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; Hadsell, R. Progressive neural networks. *arXiv* **2016**, arXiv:1606.04671.
20. He, H.; Chen, S.; Li, K.; Xu, X. Incremental learning from stream data. *IEEE Trans. Neural Netw.* **2011**, *22*, 1901–1914.
21. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
22. Buciluǎ, C.; Caruana, R.; Niculescu-Mizil, A. Model compression. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 August 2006; pp. 535–541.
23. Zhang, Y.; Xiang, T.; Hospedales, T.M.; Lu, H. Deep mutual learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4320–4328.
24. Mittal, S.; Galesso, S.; Brox, T. Essentials for class incremental learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3513–3522.
25. Castro, F.M.; Marín-Jiménez, M.J.; Guil, N.; Schmid, C.; Alahari, K. End-to-end incremental learning. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 233–248.
26. Belouadah, E.; Popescu, A. Il2m: Class incremental learning with dual memory. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 583–592.
27. Cao, G.; Cheng, Z.; Xu, Y.; Li, D.; Pu, S.; Niu, Y.; Wu, F. E2-AEN: End-to-End Incremental Learning with Adaptively Expandable Network. *arXiv* **2022**, arXiv:2207.06754.
28. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
29. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13; pp. 740–755.
30. Hao, Y.; Fu, Y.; Jiang, Y.G.; Tian, Q. An end-to-end architecture for class-incremental object detection with knowledge distillation. In Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China, 8–12 July 2019; pp. 1–6.
31. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2015; Volume 28.
32. Kwon, J.; Khalil, A.; Kim, D.; Nam, H. Incremental end-to-end learning for lateral control in autonomous driving. *IEEE Access* **2022**, *10*, 33771–33786. [CrossRef]
33. Hou, S.; Pan, X.; Loy, C.C.; Wang, Z.; Lin, D. Learning a unified classifier incrementally via rebalancing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–19 June 2019; pp. 831–839.
34. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; University of Totonto: Toronto, ON, Canada, 2009.

35. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2009; pp. 248–255.
36. Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; Fu, Y. Large scale incremental learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2019; pp. 374–382.
37. Guo, Y.; Zhang, L.; Hu, Y.; He, X.; Gao, J. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part III 14; pp. 87–102.
38. Medak, D.; Posilović, L.; Subašić, M.; Budimir, M.; Lončarić, S. Automated defect detection from ultrasonic images using deep learning. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **2021**, *68*, 3126–3134. [CrossRef] [PubMed]
39. Nawaratne, R.; Alahakoon, D.; De Silva, D.; Yu, X. Spatiotemporal anomaly detection using deep learning for real-time video surveillance. *IEEE Trans. Ind. Inform.* **2019**, *16*, 393–402. [CrossRef]
40. Kotsiantis, S.; Kanellopoulos, D.; Pintelas, P. Handling imbalanced datasets: A review. *GESTS Int. Trans. Comput. Sci. Eng.* **2006**, *30*, 25–36.
41. Chen, S.; He, H. Towards incremental learning of nonstationary imbalanced data stream: A multiple selectively recursive approach. *Evol. Syst.* **2011**, *2*, 35–50. [CrossRef]
42. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284.
43. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
44. Batista, G.E.; Prati, R.C.; Monard, M.C. A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explor. Newsl.* **2004**, *6*, 20–29. [CrossRef]
45. Tomek, I. Two modifications of CNN. *IEEE Trans. Syst. Man Cybern.* **1976**, *6*, 769–772.
46. Swana, E.F.; Doorsamy, W.; Bokoro, P. Tomek link and SMOTE approaches for machine fault classification with an imbalanced dataset. *Sensors* **2022**, *22*, 3246. [CrossRef] [PubMed]
47. Domingos, P. Metacost: A general method for making classifiers cost-sensitive. In Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 15–18 August 1999; pp. 155–164.
48. Jang, J.; Kim, Y.; Choi, K.; Suh, S. Sequential Targeting: An incremental learning approach for data imbalance in text classification. *arXiv* **2020**, arXiv:2011.10216.
49. Sahoo, D.; Pham, Q.; Lu, J.; Hoi, S.C. Online deep learning: Learning deep neural networks on the fly. *arXiv* **2017**, arXiv:1711.03705.
50. Han, H.; Yang, R.; Li, S.; Hu, R.; Li, X. SSGD: A smartphone screen glass dataset for defect detection. In Proceedings of the ICASSP 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
51. Ren, M.; Chen, N.; Qiu, H. Human-machine Collaborative Decision-making: An Evolutionary Roadmap Based on Cognitive Intelligence. *Int. J. Soc. Robot.* **2023**, *15*, 1101–1114. [CrossRef]
52. Nunes, D.S.; Zhang, P.; Silva, J.S. A survey on human-in-the-loop applications towards an internet of all. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 944–965. [CrossRef]
53. Wang, J.; Ma, Y.; Zhang, L.; Gao, R.X.; Wu, D. Deep learning for smart manufacturing: Methods and applications. *J. Manuf. Syst.* **2018**, *48*, 144–156. [CrossRef]
54. Ren, J.; Ren, R.; Green, M.; Huang, X. Defect detection from X-ray images using a three-stage deep learning algorithm. In Proceedings of the 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Halifax, NS, Canada, 5–8 May 2019; pp. 1–4.
55. Bhatt, P.M.; Malhan, R.K.; Rajendran, P.; Shah, B.C.; Thakar, S.; Yoon, Y.J.; Gupta, S.K. Image-based surface defect detection using deep learning: A review. *J. Comput. Inf. Sci. Eng.* **2021**, *21*, 040801. [CrossRef]
56. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
57. Kusiak, A. Smart manufacturing must embrace big data. *Nature* **2017**, *544*, 23–25. [CrossRef]
58. Babcock, B.; Babu, S.; Datar, M.; Motwani, R.; Widom, J. Models and issues in data stream systems. In Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, Madison, WI, USA, 3–5 June 2002; pp. 1–16.
59. Alrawashdeh, K.; Purdy, C. Toward an online anomaly intrusion detection system based on deep learning. In Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, 18–20 December 2016; pp. 195–200.
60. Shi, Z.; Li, Y.; Liu, C. Knowledge Distillation-enabled Multi-stage Incremental Learning for Online Process Monitoring in Advanced Manufacturing. In Proceedings of the 2022 IEEE International Conference on Data Mining Workshops (ICDMW), Orlando, FL, USA, 28 November–1 December 2022; pp. 860–867.
61. Lopez-Paz, D.; Ranzato, M. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; Volume 30.
62. Chaudhry, A.; Ranzato, M.; Rohrbach, M.; Elhoseiny, M. Efficient lifelong learning with a-gem. *arXiv* **2018**, arXiv:1812.00420.

63. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 3521–3526. [CrossRef] [PubMed]
64. Li, Z.; Hoiem, D. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 2935–2947. [CrossRef] [PubMed]
65. Hayes, T.L.; Kafle, K.; Shrestha, R.; Acharya, M.; Kanan, C. Remind your neural network to prevent catastrophic forgetting. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 466–483.
66. Serra, J.; Suris, D.; Miron, M.; Karatzoglou, A. Overcoming catastrophic forgetting with hard attention to the task. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 4548–4557.
67. Ramasesh, V.V.; Dyer, E.; Raghu, M. Anatomy of catastrophic forgetting: Hidden representations and task semantics. *arXiv* **2020**, arXiv:2007.07400.
68. Shmelkov, K.; Schmid, C.; Alahari, K. Incremental learning of object detectors without catastrophic forgetting. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3400–3409.
69. McCloskey, M.; Cohen, N.J. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*; Elsevier: San Diego, CA, USA, 1989; Volume 24, pp. 109–165.
70. Ramasesh, V.V.; Lewkowycz, A.; Dyer, E. Effect of scale on catastrophic forgetting in neural networks. In Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021.
71. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
72. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
73. Rebuffi, S.A.; Kolesnikov, A.; Sperl, G.; Lampert, C.H. icarl: Incremental classifier and representation learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA , 21–26 July 2017; pp. 2001–2010.
74. Rolnick, D.; Ahuja, A.; Schwarz, J.; Lillicrap, T.; Wayne, G. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2019; Volume 32.
75. Isele, D.; Cosgun, A. Selective experience replay for lifelong learning. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LO, USA, 2–7 February 2018; Volume 32.
76. Chaudhry, A.; Rohrbach, M.; Elhoseiny, M.; Ajanthan, T.; Dokania, P.K.; Torr, P.H.; Ranzato, M. On tiny episodic memories in continual learning. *arXiv* **2019**, arXiv:1902.10486.
77. De Lange, M.; Tuytelaars, T. Continual prototype evolution: Learning online from non-stationary data streams. In Proceedings of the IEEE/CVF international Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 8250–8259.
78. Shin, H.; Lee, J.K.; Kim, J.; Kim, J. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; Volume 30.
79. Atkinson, C.; McCane, B.; Szymanski, L.; Robins, A. Pseudo-recursal: Solving the catastrophic forgetting problem in deep neural networks. *arXiv* **2018**, arXiv:1802.03875.
80. Lavda, F.; Ramapuram, J.; Gregorova, M.; Kalousis, A. Continual classification learning using generative models. *arXiv* **2018**, arXiv:1810.10612.
81. Ramapuram, J.; Gregorova, M.; Kalousis, A. Lifelong generative modeling. *Neurocomputing* **2020**, *404*, 381–400. [CrossRef]
82. Aljundi, R.; Lin, M.; Goujaud, B.; Bengio, Y. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2019; Volume 32.
83. Lee, S.W.; Kim, J.H.; Jun, J.; Ha, J.W.; Zhang, B.T. Overcoming catastrophic forgetting by incremental moment matching. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; Volume 30.
84. Zenke, F.; Poole, B.; Ganguli, S. Continual learning through synaptic intelligence. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, NSW, Australia, 6–11 August 2017; pp. 3987–3995.
85. Liu, X.; Masana, M.; Herranz, L.; Van de Weijer, J.; Lopez, A.M.; Bagdanov, A.D. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 2262–2268.
86. Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; Tuytelaars, T. Memory aware synapses: Learning what (not) to forget. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 139–154.
87. Chaudhry, A.; Dokania, P.K.; Ajanthan, T.; Torr, P.H. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2018; pp. 532–547.
88. Jung, H.; Ju, J.; Jung, M.; Kim, J. Less-forgetting learning in deep neural networks. *arXiv* **2016**, arXiv:1607.00122.
89. Rannen, A.; Aljundi, R.; Blaschko, M.B.; Tuytelaars, T. Encoder based lifelong learning. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1320–1328.
90. Zhang, J.; Zhang, J.; Ghosh, S.; Li, D.; Tasci, S.; Heck, L.; Zhang, H.; Kuo, C.C.J. Class-incremental learning via deep model consolidation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 1131–1140.
91. Mallya, A.; Lazebnik, S. Packnet: Adding multiple tasks to a single network by iterative pruning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7765–7773.

92. Fernando, C.; Banarse, D.; Blundell, C.; Zwols, Y.; Ha, D.; Rusu, A.A.; Pritzel, A.; Wierstra, D. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv* **2017**, arXiv:1701.08734.
93. Mallya, A.; Davis, D.; Lazebnik, S. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 67–82.
94. Aljundi, R.; Chakravarty, P.; Tuytelaars, T. Expert gate: Lifelong learning with a network of experts. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3366–3375.
95. Xu, J.; Zhu, Z. Reinforced continual learning. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2018; Volume 31.
96. Rosenfeld, A.; Tsotsos, J.K. Incremental learning through deep adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *42*, 651–663. [CrossRef] [PubMed]
97. Xu, M.; Zhao, Y.; Liang, Y.; Ma, X. Hyperspectral Image Classification Based on Class-Incremental Learning with Knowledge Distillation. *Remote Sens.* **2022**, *14*, 2556. [CrossRef]
98. Yoon, J.; Yang, E.; Lee, J.; Hwang, S.J. Lifelong learning with dynamically expandable networks. *arXiv* **2017**, arXiv:1708.01547.
99. Hou, S.; Pan, X.; Loy, C.C.; Wang, Z.; Lin, D. Lifelong learning via progressive distillation and retrospection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 437–452.
100. Ostapenko, O.; Puscas, M.; Klein, T.; Jahnichen, P.; Nabi, M. Learning to remember: A synaptic plasticity driven framework for continual learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 11321–11329.
101. Li, D.; Tasci, S.; Ghosh, S.; Zhu, J.; Zhang, J.; Heck, L. RILOD: Near real-time incremental learning for object detection at the edge. In Proceedings of the 4th ACM/IEEE Symposium on Edge Computing, Arlington, VA, USA, 7–9 November 2019; pp. 113–126.
102. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
103. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; pp. 21–37.
104. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
105. Belouadah, E.; Popescu, A. DeeSIL: Deep-Shallow Incremental Learning. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
106. Sudharsan, B.; Yadav, P.; Breslin, J.G.; Ali, M.I. Train++: An incremental ml model training algorithm to create self-learning iot devices. In Proceedings of the 2021 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI), Atlanta, GA, USA, 18–21 October 2021; pp. 97–106.
107. Qin, Z.; Yu, F.; Chen, X. Task-adaptive incremental learning for intelligent edge devices. *arXiv* **2019**, arXiv:1910.03122.
108. Hussain, M.A.; Huang, S.A.; Tsai, T.H. Learning with sharing: An edge-optimized incremental learning method for deep neural networks. *IEEE Trans. Emerg. Top. Comput.* **2022**, *11*, 461–473. [CrossRef]
109. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
110. Mohandas, R.; Bhattacharya, M.; Penica, M.; Camp, K.V.; Hayes, M.J. TensorFlow Enabled Deep Learning Model Optimization for enhanced Realtime Person Detection using Raspberry Pi operating at the Edge. In Proceedings of the 28th Irish Conference on Artificial Intelligence and Cognitive Science, Dublin, Ireland, 7–8 December 2020; Volume 2771, pp. 157–168.
111. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 886–893.
112. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
113. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
114. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
115. Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; Zhang, G. Learning under concept drift: A review. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 2346–2363. [CrossRef]
116. Yu, Y.Y.; Zhang, D.; Zhang, X.M.; Peng, X.B.; Ding, H. Online stability boundary drifting prediction in milling process: An incremental learning approach. *Mech. Syst. Signal Process.* **2022**, *173*, 109062. [CrossRef]
117. Li, J.; Dai, Q.; Ye, R. A novel double incremental learning algorithm for time series prediction. *Neural Comput. Appl.* **2019**, *31*, 6055–6077. [CrossRef]
118. Camargo, E.; Aguilar, J.; Quintero, Y.; Rivas, F.; Ardila, D. An incremental learning approach to prediction models of SEIRD variables in the context of the COVID-19 pandemic. *Health Technol.* **2022**, *12*, 867–877. [CrossRef] [PubMed]

119. Pierre, J.M. Incremental lifelong deep learning for autonomous vehicles. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 3949–3954.
120. Ramos, D.; Faria, P.; Vale, Z.; Mourinho, J.; Correia, R. Industrial facility electricity consumption forecast using artificial neural networks and incremental learning. *Energies* **2020**, *13*, 4774. [CrossRef]
121. Yu, W.; Zhao, C. Broad convolutional neural network based industrial process fault diagnosis with incremental learning capability. *IEEE Trans. Ind. Electron.* **2019**, *67*, 5081–5091. [CrossRef]
122. Zizic, M.C.; Mladineo, M.; Gjeldum, N.; Celent, L. From industry 4.0 towards industry 5.0: A review and analysis of paradigm shift for the people, organization and technology. *Energies* **2022**, *15*, 5221. [CrossRef]
123. Alsamhi, S.H.; Shvetsov, A.V.; Kumar, S.; Hassan, J.; Alhartomi, M.A.; Shvetsova, S.V.; Sahal, R.; Hawbani, A. Computing in the sky: A survey on intelligent ubiquitous computing for uav-assisted 6g networks and industry 4.0/5.0. *Drones* **2022**, *6*, 177. [CrossRef]
124. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
125. Sarwar, S.S.; Ankit, A.; Roy, K. Incremental learning in deep convolutional neural networks using partial network sharing. *IEEE Access* **2019**, *8*, 4615–4628. [CrossRef]
126. Kalal, Z.; Matas, J.; Mikolajczyk, K. Online learning of robust object detectors during unstable tracking. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), Kyoto, Japan, 27 September–4 October 2009; pp. 1417–1424.
127. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2012; Volume 25.
128. Kuznetsova, A.; Rom, H.; Alldrin, N.; Uijlings, J.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Kolesnikov, A.; et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *Int. J. Comput. Vis.* **2020**, *128*, 1956–1981. [CrossRef]
129. Sun, C.; Shrivastava, A.; Singh, S.; Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 843–852.
130. Lomonaco, V.; Maltoni, D. Core50: A new dataset and benchmark for continuous object recognition. In Proceedings of the Conference on Robot Learning, PMLR, Mountain View, CA, USA, 13–15 November 2017; pp. 17–26.
131. Li, H.; Singh, B.; Najibi, M.; Wu, Z.; Davis, L.S. An analysis of pre-training on object detection. *arXiv* **2019**, arXiv:1904.05871.
132. Hendrycks, D.; Lee, K.; Mazeika, M. Using pre-training can improve model robustness and uncertainty. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 2712–2721.
133. Zoph, B.; Ghiasi, G.; Lin, T.Y.; Cui, Y.; Liu, H.; Cubuk, E.D.; Le, Q. Rethinking pre-training and self-training. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2020; Volume 33, pp. 3833–3845.
134. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
135. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747.
136. Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A.Y. Reading digits in natural images with unsupervised feature learning. In Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning, Granada, Spain, 12–17 December 2011.
137. Krizhevsky, A.; Nair, V.; Hinton, G. The CIFAR-10 Dataset. 2014. Available online: https://www.cs.toronto.edu/~kriz/cifar.html (accessed on 17 October 2023).
138. Le, Y.; Yang, X. Tiny imagenet visual recognition challenge. *CS 231N* **2015**, *7*, 3.