Fine-Grained Change Detection

Fine-grained change detection in sensor data is very challenging for artificial intelligence though it is critically important in practice. It is the process of identifying differences in the state of an object or phenomenon where the differences are class-specific and are difficult to generalise. As a result, many recent technologies that leverage big data and deep learning struggle with this task.

1. Introduction

Change detection (CD), the process of identifying differences in object/phenomena over time/space, is often considered a fundamental low-level preprocessing step in many data analysis problems, such as in sensor data analytics, computer vision and process trend analysis. However, it can also be considered the primary task in many real-world applications such as remote sensing, surveillance, security and healthcare. The major challenge of CD is to separate real changes from false changes caused by different sensing conditions, e.g., sensor noise, suddenly varied lightings and camera movements in computer vision and unexpected changes in data distributions.

Most state-of-the-art CD methods assume real changes occur on a relatively large amount of data and are salient enough to transcend detailed changes caused by these factors. However, there are many applications where it is not feasible to collect data of sufficient breadth or depth for this method to be reliable, i.e., interactions between different combinations of conditions that were not accounted for at the design stage can induce variability that clouds and alters the characteristic features of significant changes, especially to each scenario. Clearly, for such scenarios, it is difficult for even the most modern deep learning techniques to generalise the features of changes of interest. This article will review the current state-of-the-art methods and some of the challenges to reliable detection of fine-grained change. In particular, we focus on techniques that can be applied to the representations learned by artificial intelligence in multi-task, multi-modal, open-set and online learning settings with little data to aid in navigating variability and uncertainty so that significant changes become apparent.

RL refers to the methodology of learning to represent data in the most simple form possible that preserves the details relevant to the task(s) at hand. RL is an integral part of many machine learning algorithms and comes in different guises but all essentially have the common goal of defining a feature space in which we can make observations on the relation between entities.

2. Applications of Change Detection

Change detection is quite a broad term that encapsulates anything from low-level processes in algorithms such as edge detection to high-level tasks that must employ contextual understanding to determine significant change. This section will review applications of the latter, which include methods for detecting differences on a spatial scale, on a time scale, on triggered objects or on some hybrid of these types.

In many of these applications, it is sometimes desirable to distinguish instances of change by capturing slight and subtle differences. For instance, it may be desirable to track the trend of continuous change in the recent past (e.g., to track the progression of a disease) for each instance. It is also often necessary to accommodate intra-class variation for a CD system to be effective in its intended application, i.e., in applications such as biomedical diagnosis and all-important buildings (e.g., dam) monitoring, it is critical to guarantee detection sensitivity and accuracy of minute changes in each observation by taking measures to maximise the signal-to-noise ratio by adapting our reasoning specific to the class of object we are looking at.

This practice is known as fine-grained (FG) data analysis, which targets the study of objects/phenomena from subordinate categories, e.g., if the base task is to detect changes in human health, the FG task may be to detect changes specific to a specific person. FG analysis is a long-standing and fundamental problem because small inter-class variations in the phenomenon of interest can often be masked by large intra-class variations de to ancillary data [1]. However, it is an important problem and has become ubiquitous in diverse CD applications such as automatic biodiversity monitoring [2], climate change evaluation [3], intelligent retail [4], intelligent transportation [5], and many more.

Remote sensing (RS) is the collection of images of an object/area from afar, typically from a satellite or aircraft and usually of the Earth's surface. CD is an important aspect of RS as a tool to reliably quantify spectral differences in the radiation received from features of interest, whether it be for the study of spatial differences in surveying applications such as land use and land cover classification [6], agricultural analyses [7], environmental monitoring [3], disaster assessment [8] and map revision [9].

Handling uncertainty is one of the main concerns in these applications as many external factors, such as sensor gain (random error due to imperfect calibrated camera sensor arrays), image noise and atmospheric conditions [10] influence the absolute sensor readings, which means that corresponding subtle differences between images, even in the same location, in the large datasets, which are typically accrued, is not so straightforward. Specialised CD techniques for addressing this concern include fuzzy logic, Monte Carlo analysis and geostatistical analysis [11].

Fuzzy logic employs membership functions to express the vagueness of labels (e.g., land cover may vary continuously in transition zones), thus fuzzy classes are assigned in proportion for each entity and some ambiguity is mitigated. Uncertainty due to human error during the manual labelling has also been taken into account by explicitly incorporating label jitter (inconsistencies in labelling near class boundaries arising from human error in the annotation process) into the model training process in the form of an activity boundary smoothing method that explicitly allows overlapping activity labels [10]. The Monte Carlo method is a paradigm that has to do with randomness—a random sample, drawn from the error probability distribution of each measurement, is added to that measurement, and the net effect on the overall picture is stored. This procedure is repeated several hundred times and the resulting collection of maps is analysed to see how measurement uncertainty has propagated to the outcome. If many of the maps show a large variation in a measurement at a particular location, then we know there is a lot of uncertainty. Lastly, geostatistics can also be useful in improving measurements in remote sensing through the use of statistical understandings of spatially varying properties.

Terrestrial based mapping applications also apply such CD techniques to overcome uncertainty arising from large sudden changes in camera pose, dynamic objects (i.e., objects that can be removed from a scene and thereby affect its appearance) and limited field of view. Three-dimensional sensing has become very popular for aiding in overcoming some of these challenges as recently, sensors have become available that can provide reliable depth information for each pixel. These sensors allow the physical geometry of objects to be measured with relative immunity to illumination variations and perspective distortions, which enables simple geometric comparisons of extracted 3D shapes with simulated reference shapes to be effective for change detection [12]. Challenges in this area include misalignment in point cloud registration and designing algorithms efficient enough to compensate for the increasing data volume.

In simple computer vision applications, where the sources of uncertainty can be constrained (e.g., in industrial manufacturing lines where lighting and environmental conditions are well controlled), CD techniques such as edge detection in images are a powerful tool. For example, high precision industrial vision/sensing systems for the inspection and categorisation of objects can achieve accuracies well within the allowable tolerance of standard measurement instruments automatically, non-invasively and without requiring precise fixturing with the aid of high-resolution cameras, a lot of specialised knowledge in machine vision and edge detection [13] and sub-pixel detection techniques [14].

The most common use cases of more complex applications of CD in video surveillance to date entail abnormal changes of foreground human behaviours/activities that could pose damage or danger to human properties and lives, e.g., fall detection [15], aggressive/violent behaviour detection [16] and pedestrian intention estimation for advanced driver-assistance systems (ADAS) [17]. These applications require change-detection to happen in real-time and in unregulated environments (environments where variables such as lighting conditions, camera pose, object pose and object characteristics are relatively ill-constrained compared to industrial/laboratory conditions). The challenges associated with these requirements are discussed further in Section 5.1.

CD is an extremely common task in the healthcare sector since medical diagnoses are essentially based on the difference between a patient's state and known "healthy" conditions or their previous state. Scientists are now trying to automate some of these processes to relieve some of the burden on the medical sector arising from the demographically older population and enable more ubiquitous and personalised remote healthcare solutions. Some of this research investigates

the use of wireless sensors for monitoring the physiological profile of the wearer in a continuous, real-time, and non-intrusive manner for the early detection of illness/incident [18][19]. Continuous monitoring involves the recognition of complex patterns across a wide variety of scenarios, e.g., as patients make lifestyle changes during recovery, and fine-grained analysis as each patient will behave differently [20]. It is also desirable to perform CD on the edge (i.e., for the algorithms to be processed on or close to the sensor in an Internet of Things network) to mitigate the need for raw data to be transmitted and save bandwidth but more importantly where real-time data processing and decision making are important for closed-loop systems that must maintain critical physiological parameters [21]. The reduced processing and memory capability of hardware on the edge necessitates for algorithms to be lightweight and efficient. Maintaining CD performance in the face of problems deriving from changes in data distribution over time is also a challenge for which distributed learning systems are a promising proposition.

CD algorithms also play an important role in diagnostic fields involving signal analysis such as cardiology [22] and the analysis of medical images, e.g., in retinopathy and radiography [23]. CD also has applications in sensor-assisted/robot-assisted surgery in the analysis of data from sensors for detecting changes in tissue characteristics [24].

Complex computer-based systems aimed to assist/automate tasks that consist of multiple interconnected components take considerable effort to maintain. The monitoring and alerting of changes to the procedures within these systems is of great importance to ensure no alterations made during system maintenance interfere with critical functions. Examples where CD has been implemented include clinical decision support systems [25], web ontologies [23] and safety-critical software [26].

The modelling of dynamic systems can also be considered an application of CD principles, e.g., in the detection of sensor and actuator failures [27] and the tracking of manoeuvring vehicles/robots [28]. System dynamics endeavours to derive a mathematical model of the non-linear behaviour of complex systems in order to understand and track them effectively. In practice, these models not only have to reflect the behaviour of the system but must also accommodate deficiencies in the sensing hardware used to monitor it. For example, some models account for measurement drift by appending a second-order term that describes the characteristic behaviour of the sensor between calibrations [29] while others learn the interaction between the system and sensor(s) as a whole with a neural network [27]. In addition, abrupt sensor faults can be addressed by sampling over a longer time window when training such a neural network [27].

3. History of Change Detection

In this section, we will give a brief overview of the evolution of the tools available in the field of CD. As these tools progressed, the size, dimensionality and complexity of the data the algorithms were capable of processing also progressed. Methods initially focused on univariate time series data that followed parametric assumptions and then began learning non-linear relationships in non-parametric sequential data with machine learning, eventually being able to model multivariate, non-stationary data and finally were able to process high-dimensional computer vision data with deep learning.

Early research in CD was concerned with change point detection in sequential data. The main application area for this research was industrial statistical process control (SPC), where the approach is to detect the changes in the mean of the time series, assuming the baseline process to be stationary and the shift pattern to be a step function that is sustained after the shift. The theory behind change point detection is known as sequential analysis. Some notable methods include Seasonal Trend Decomposition using LOESS (Locally Estimated Scatterplot Smoothing) [30] and PELT (Pruned Exact Linear Time) algorithm [31]. STL decomposes the time series into three components: trend, season and residual where the rate of change and smoothness of the season and trend, respectively, can be tuned to the periodicity of the input data.

Slightly more powerful statistical CD schemes for non-parametric problems are based on generalised likelihood ratio statistics [32], which assume that signal patterns follow a known distribution during "normal" conditions and deviation from this distribution is distinguishable and is an indicator that a change has occurred. These methods are far more "automatic" in that they do not require manual oversight or tuning. A classic example is the Conventional Cumulative Sum (CUSUM) algorithm, which monitors the correlation of signal patterns with, for example, a Gaussian distribution with mean and known standard deviation , and accumulates deviations from these statistics until they reach a certain threshold. If the threshold is reached within a predefined time window then a change has been detected [33]. Some variants of CUSUM are also able to handle non-stationary sequences (where the "normal" distribution can shift) [34] and FG risk adjustment (by replacing static control limits with simulation-based dynamic probability control limits for each subject) [35].

In applications where data may be subject to a variety of sources of variation that influence the distribution of occurrence of particular phenomena (e.g., long-term periodic signal variation due to the day of the week/time of day, etc.), the source of deviations may be accounted for and recognised so as not to falsely trigger real anomalies. However, models become increasingly complex the more exclusions it has to accommodate and it is often not possible to identify all possible sources of noise during system design. Therefore, algorithms must be able to automatically learn to differentiate noise from natural signal variation in a wide variety of scenarios with limited information. This class of algorithm is known as machine learning, of which early methods used techniques such as Gaussian Mixture Models, which represent signal relations as probability distributions and compare them against each other [35], or kernel functions and later work, which took advantage of the acceleration of machine learning with parallel processing, which we will cover in the next section.

Recently, there has been a big jump in our ability to recognise complex features thanks to a development called deep learning (DL), and more specifically, the neural network (NN) computing architecture, which emulates the theorised functioning of the human brain. The adjective "deep" is often assumed to mean that the architecture consists of many layers of computing cells, sometimes called "neurons", that each perform a simple operation. The result of each computation being an activation signal that is passed through to the neurons in proceeding layers. Each neuron assigns a weight to each of its inputs and adds a bias value if necessary. By tuning these weights and biases, a model can be trained/learned to capture "deeper" local information and features through exploiting self-organisation and interaction between small units [36]. It is also for this reason that deep neural networks (DNNs) are often computed using GPUs, or similar hardware suited to matrix multiplication, and the availability of such computing resources is what has fuelled the recent activity and great strides in the predictive capability of artificial intelligence.

The power of DL comes at the cost of the need for large amounts of data to learn from. In terms of whether this data requires manual labels, most deep learning approaches can be grouped into supervised and unsupervised methods. Supervised methods can generalise better but only where large annotated datasets are available, which for less popular applications such as CD and FG recognition is not that common. However, there are many methods for training DL models in such circumstances, in both supervised and unsupervised settings [9], including one-shot learning, generative-adversarial learning and structure/theory-based methods. These topics may be considered to be forms of Representation Learning, a division of DL where the emphasis is on encoding higher-order statistics of convolutional activations/features learnt by a DNN to enhance the mid-level learning capability, i.e., the focus is on enhancing the intermediate feature descriptor learned by a DL model to output a "good" representation of the input data.

Generating "good" representations entails providing a means of discrimination based on intrinsic data properties while also determining the relation between entities. Hence, progress in this field is naturally applicable to FG CD applications. To summarise briefly, the three aforementioned types of Representation Learning lend themselves to different types of tasks. The most common form of one-shot learning, metric learning, is most suited to tasks where a definite global reference metric is available for the output to be predicted, i.e. supervised change detection. Whereas generative methods are more suited to discovering intrinsic patterns in data and displaying these patterns such that significant changes become apparent. The third type, structure/theory-based representation learning, primarily involves using graph theory, i.e. Geometric Deep Learning. GDL combines the best of both worlds in being able to learn in situations where we have information available on the relationship between entities while also in being able to construct these graphs in an unsupervised manner.

4. Challenges, Comparisons, and Future Directions for Change Representation Techniques

The previous section details a number of techniques that have arisen from a diverse range of application domains to address challenges and leverage opportunities often specific to the traits of the data available/requirements of the application. In this section, we group some of these challenges under categories relating to requirements for adaptable real-time response, input data inconsistencies and model interpretability. Under each category, we discuss some recent approaches to these problems and offer some perspectives on trends in the uptake of some of these techniques towards addressing these problems.

Most CD applications require change detection to be performed in real-time, i.e., they require data to be processed sequentially and for change-points to be detected as soon as they occur or within a certain time window [37]. This can be considerably more challenging as retrospective offline techniques have the advantage of access to the data before and after the point to decide whether the data distribution has changed. This problem is known as quickest change detection (QCD) [15] and is common in applications such as manufacturing quality control and fall/incident detection in patient monitoring. Furthermore, these applications typically require the algorithms to be deployable on edge devices, which implies real-time processing with limited computation complexity. The more basic statistical methods excel in terms of

computation time and hence are still relevant if the problem is not too complex, e.g., seasonal-trend decomposition and likelihood ratio statistics to detect the changes [37]. The segmentation approach used in graphical methods suffers here due to the high dimensionality of the output difference image/change map; although, real-time detection is possible if trained properly.

Another related field of research that deals with the challenge of applying deep learning to data on the fly is online learning, which requires new classes to be recognised at deployment. Continual learning or lifelong learning refers to the ability to continually learn over time by accommodating new knowledge while retaining previously learned experiences [38] [39]. The catastrophic forgetting problem, mentioned in Section 4.2.1, is present, and with regards to FGCD, we identify the process of CD as being a key tool for continual learning in general. It has been demonstrated by [40] that detecting changes in dense RGB-D maps over the lifetime of a robot can aid in automatically learning segmentations of objects.

There are many challenges associated with heterogeneous data sources, i.e., the input data for each of the tasks might contain missing values, the scale and resolution of the values is not consistent across tasks and the data contain non-IID instances.

A methodology that may be applied to non-visual data/a hybrid of visual and non-visual data is to first convert the non-visual data so that it can be viewed as an image (e.g., activity data from wearable sensors can be visualised in the form of a density map that uses different colours to show varying levels of activity [41][42]) and then proceed with image-based techniques. However, the way that the data are encoded into image form can influence the results as most convolution-based networks are not permutation invariant.

Another technique that is useful for continuous variables is kernelisation, which is a technique for replacing input with a kernel, a function that is symmetric and positive definite. By virtue of positive-definiteness, the kernel function allows us to transform our input to a domain where we can solve problems more-efficiently and then use tricks discovered in that domain in the original domain. A classic example of this is in use in support vector machines for non-linear regression. Furthermore, kernelisation can allow us to represent the desired output on ordinal, interval or ratio scales, which may be more useful in some applications. A number of papers have proposed techniques for performing regression with DML using kernelisation [43][44][45].

Sparse compositional metric learning was proposed by [46]. It learns local Mahalanobis metrics for multi-task/multi-class data on sparse combinations of rank-one basis metrics. Sparse metric learning pursues dimension reduction and sparse representations during the learning process using mixed-norm regularisation, which results in much faster and efficient distance calculation [47]. This concept also allows learning on sparse and unbalanced data. Much of this type of research took place before the advent of deep learning, and therefore, there is an opportunity for these techniques to be applied to deep networks.

Explainable artificial intelligence (XAI) refers to AI that produces details or reasons to make its functioning clear or easy to understand. These principles can be applied to the interpretation of latent spaces in RL to assist the evaluation of models, help explain model performance, and more generally aid understanding of what exactly a model has "learned" [48].

For example, some papers use discriminative clustering in latent spaces to decide whether different classes form distinct clusters; however, if we want to explore the latent space further to understand the underlying structures in the data, we need visualisation tools [48]. From these analyses, one may discover useful metrics that may be exploited, e.g., clusters in the latent space may be found to reflect that distance between the same words from embeddings trained on different corpora signifies a change in word meaning in certain contexts [49].

A key decision to be made when interpreting latent space, or indeed during any data analysis, is whether the identified features represent true features of the underlying space rather than artefacts of sampling. A common example of misreading projections of latent space is with t-SNE, where conclusions are drawn without trialling different parameters of the projection algorithm such as the perplexity that needs to be tuned in proportion to approximately the number of close neighbours each point has in order to balance attention between local and global aspects of the data.

Persistent homology (PH) is a method for automating this type of procedure by computing the topological features of a space at different spatial resolutions. [50]. Topology provides a set of natural tools that, amongst other things, allows the intrinsic shape of the data to be detected using a provided distance. As well as being integral to geometric deep learning, the field of research known as topological data analysis (TDA) has gained popularity in recent years using these tools to quantify shape and structure in data to answer questions from the data's domain [51].

While homology measures the structure of a single, stagnant space, persistent homology watches how this structure changes as the space changes. Each data point is plotted on a persistence diagram as a pair of numbers (*a*,*b*) corresponding to its birth diameter and death diameter (i.e., the test instances at which a feature was first seen and last seen). More persistent features appear far away from the diagonal on a persistence diagram, are detected over a range of spatial scales and are deemed less likely to be due to noise or a particular choice of parameters. Persistent homology is just one form of topological signature that can show a great deal of information about a set of data points such as clustering without expert-chosen connectivity parameters and loops and voids that are otherwise invisible [51].

Once a change is detected and determined significant, additional analyses are required to explain the reason change that occurred. This problem is formally known as change analysis (CA), a method of examination beyond CD to explain the nature of discrepancy [52]. This field of research has explored methods for detecting and explaining change in time series data [53], remote sensing data [54] and diagnosis prediction. CA methods can be classified as being parametric or non-parametric. The former is where a parametric functional form is explicitly assumed to model the distribution.

For example, state-of-the-art methods learn to identify discriminative parts from images of FG categories through the use of methods for interpreting the layers of convolutional neural networks, e.g., Grad-CAM (gradient-weighted class activation mapping) [55] and LIME (local interpretable model-agnostic explanations) [56]. However, the power of these methods is limited when only few training samples are available for each category. To break this limit, possible solutions include identifying auxiliary data that are more useful for change detection specific to each class and also better at leveraging these auxiliary data [57]. Recently, there has been some interesting progress in applying Grad-CAM techniques to metric-learnt representations by [58], who generate point-to-point activation intensity maps between query and retrieve images to show the relative contribution of the different regions to the overall similarity. Not only can this technique produce better activation maps, but they are also instance-specific, which we believe is ground-breaking for FG analyses.

The incorporation of causal reasoning into ML research has also been gaining popularity in recent years. Traditionally, focusing on probabilities and correlation, ML and statistics generally avoid reasoning about cause and effect. However, this teaching has been criticised as being detrimental to the potential understanding, which can be gained from techniques such as counterfactual explanations, a specific class of explanation that provides a link between what could have happened had input to a model been changed in a particular way [59].

Theoretical research interests related to modelling complex systems require, not only for system dynamics to be captured and detected by a model but also for these changes to fit with what we currently understand about the system, e.g., to comply with the equations we have derived. Incorporating domain knowledge can be hugely advantageous as the theoretical model provides guidance with which an effective model is supposed to follow; it helps an optimised solution to be more stable and avoid over-fitting, it allows training with less data, it would be more robust to unseen data, and thus it is easier to be extended to applications with changing distributions [60]. However, this type of approach is only applicable to problems that have been studied extensively, as explaining the origin of change in terms of individual variables is generally a tough task unless the variables are independent.

Applications where theoretically grounded CD has been implemented include climate change [61] and dynamic systems [10]. These works implement techniques related to knowledge injection discussed in <u>Section 5.3.4</u>. Generally, they use an architecture based on graph networks to incorporate prior knowledge given as a form of partial differential equations (PDEs) over time and space. These PDEs can comprise very sophisticated mathematics, e.g., Lagrangian [62] and Hamiltonian mechanics [63].

Latent space visualisations can seem arbitrary and not very meaningful when the dimensions of projections of the latent space are not aligned/scaled to important metrics specific to the application.

The performance of the RL crucially determines the type and performance of the algorithm for delineating the separation between feature sets to a manageable number of dimensions. However, techniques such as sparse metric learning can also be applied to further reduce the dimensionality of the embedding representation. Methods for sparse metric learning include mixed-norm regularisation across various learning settings to whittle down latent dimensions that do not consistently contribute to producing distinguishable representations [47] and sparse compositional metric learning, which learns local Mahalanobis metrics on sparse combinations of rank-one basis metrics [46].

Expressing representations in relation to familiar metrics can be useful in the visual evaluation of model performance by highlighting cases where there was an underlying pattern not explained by the primary tasks (e.g., scene change detection) of an RL approach but due to some other ancillary variables (e.g., weather). This may be applied to RL to reveal the interactions of background/ancillary variables by these variables to the axes of latent space/manifold

visualisations, i.e., it may be useful to be able to tell why an object was classified to belong to a particular sub-class through observation of where that object lies on a space projection. We propose that by using interactive latent space cartography, which allows custom axes and colours according to selectable variables of interest, such relationships may become easily revealed. Moreover, it will help make the resulting visualisation of the embedding space more meaningful for the application. Such a visualisation of the feature space that takes into account known priors (e.g., weather conditions) has been shown to be useful in further refining the predictions at runtime [57].

If such auxiliary variables are known before inference, it may also be useful to narrow down the CD results to instances that are more likely in light of this new knowledge. This is known as knowledge injection and has been implemented in different ways depending on the type of RL. Auxiliary knowledge can be encoded as sparse input to metric learning techniques, as rules for more accurate relation extraction in generative approaches [64], or to predict missing links in knowledge graphs [65][66]. Alternatively, a clustering algorithm, e.g., k-means clustering, could be formulated taking as input the salient background variables and outputting a function that maps the latent space to valid classifications, thus maximising the inter-class variance in FG applications.

References

1. Wei, X.S.; Wu, J.; Cui, Q. Deep learning for fine-grained image analysis: A survey. arXiv 2019, arXiv:1907.03069.

2. Mallet, C.; Le Bris, A. Current challenges in operational very high resolution land-cover mapping. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 2020, 43, 703–710.

3. Lahoz, W.A.; Schneider, P. Data assimilation: Making sense of Earth Observation. Front. Environ. Sci. 2014, 2, 16.

4. Paolanti, M.; Pietrini, R.; Mancini, A.; Frontoni, E.; Zingaretti, P. Deep understanding of shopper behaviours and interactions using RGB-D vision. Mach. Vis. Appl. 2020.

5. Guerrero-Ibáñez, J.; Zeadally, S.; Contreras-Castillo, J. Sensor technologies for intelligent transportation systems. Sensors 2018, 18, 1212.

6. Ziemann, A.K.; Ren, C.X.; Theiler, J. Multi-sensor anomalous change detection at scale. In Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imagery XXV; Messinger, D.W., Velez-Reyes, M., Eds.; SPIE: Baltimore, MD, USA, 2019; Volume 10986, p. 37.

7. Awty-Carroll, K.; Bunting, P.; Hardy, A.; Bell, G. An Evaluation and Comparison of Four Dense Time Series Change Detection Methods Using Simulated Data. Remote Sens. 2019, 11, 2779.

8. Qin, D.; Zhou, X.; Zhou, W.; Huang, G.; Ren, Y.; Horan, B.; He, J.; Kito, N. MSIM: A change detection framework for damage assessment in natural disasters. Expert Syst. Appl. 2018, 97, 372–383.

9. Shi, W.; Zhang, M.; Zhang, R.; Chen, S.; Zhan, Z. Change detection based on artificial intelligence: State-of-the-art and challenges. Remote Sens. 2020, 12, 1688.

10. Senanayake, R.; Ott, L.; O'Callaghan, S.; Ramos, F. Spatio-temporal hilbert maps for continuous occupancy representation in dynamic environments. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Kyoto, Japan, 16–21 October 2016; pp. 3925–3933.

11. Foody, G.M.; Atkinson, P.M. Uncertainty in Remote Sensing and GIS; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2002; pp. 1–307.

12. Qin, R.; Tian, J.; Reinartz, P. 3D change detection—Approaches and applications. ISPRS J. Photogramm. Remote. Sens. 2016, 122, 41–56.

13. Lopez-Molina, C.; De Baets, B.; Bustince, H. Quantitative error measures for edge detection. Pattern Recognit. 2013, 46, 1125–1139.

14. Xie, X.; Ge, S.; Xie, M.; Hu, F.; Jiang, N. An improved industrial sub-pixel edge detection algorithm based on coarse and precise location. J. Ambient Intell. Humaniz. Comput. 2020, 11, 2061–2070.

15. Tao, J.; Turjo, M.; Tan, Y.P. Quickest change detection for health-care video surveillance. In Proceedings of the 2006 IEEE International Symposium on Circuits and Systems, Kos, Greece, 21–24 May 2006; pp. 505–508.

16. Gao, Y.; Liu, H.; Sun, X.; Wang, C.; Liu, Y. Violence detection using Oriented VIolent Flows. Image Vis. Comput. 2016, 48, 37–41.

17. Kataoka, H.; Satoh, Y.; Aoki, Y.; Oikawa, S.; Matsui, Y. Temporal and fine-grained pedestrian action recognition on driving recorder database. Sensors 2018, 18, 627.

18. Aslam Khan, F.; Member, S.; Hasan Haldar, N.A.; Ali, A.; Iftikhar, M.; Zia, T.A.; Zomaya, A.Y. A Continuous Change Detection Mechanism to Identify Anomalies in ECG Signals for WBAN-Based Healthcare Environments. IEEE Access 2017, 5, 13531–13544.

19. Riboni, D.; Bettini, C.; Civitarese, G.; Janjua, Z.H.; Helaoui, R. SmartFABER: Recognizing Fine-grained Abnormal Behaviors for Early Detection of Mild Cognitive Impairment. Artif. Intell. Med. 2016, 67, 57–74.

20. Sprint, G.; Cook, D.J.; Schmitter-Edgecombe, M. Unsupervised detection and analysis of changes in everyday physical activity data. J. Biomed. Inform. 2016, 63, 54–65.

21. Satija, U.; Ramkumar, B.; Manikandan, M.S. Robust cardiac event change detection method for long-term healthcare monitoring applications. Healthc. Technol. Lett. 2016, 3, 116–123.

22. Colt, R.G.; Várady, C.H.; Volpi, R.; Malagò, L. Automatic Feature Extraction for Heartbeat Anomaly Detection. arXiv 2021, arXiv:2102.12289.

23. Klein, M.; Fensel, D.; Kiryakov, A.; Ognyanov, D. Ontology versioning and change detection on the web. In Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin/Heidelberg, Germany, 2002; Volume 2473, pp. 197–212.

24. Uribe, D.O.; Schoukens, J.; Stroop, R. Improved Tactile Resonance Sensor for Robotic Assisted Surgery David. Mech. Syst. Signal Process. 2018, 99, 600–610.

25. Liu, S.; Wright, A.; Hauskrecht, M. Change-point detection method for clinical decision support system rule monitoring. In Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin/Heidelberg, Germany, 2017; Volume 10259, pp. 126–135.

26. Borg, M.; De La Vara, J.L.; Wnuk, K. Practitioners' Perspectives on Change Impact Analysis for Safety-Critical Software —A Preliminary Analysis. In International Conference on Computer Safety, Reliability, and Security; Springer: Cham, Switzerland, 2016; pp. 346–358.

27. Patel, H.R.; Shah, V.A. Passive Fault Tolerant Control System Using Feed-forward Neural Network for Two-Tank Interacting Conical Level Control System against Partial Actuator Failures and Disturbances. In IFAC-PapersOnLine; Elsevier B.V.: Amsterdam, The Netherlands, 2019; Volume 52, pp. 141–146.

28. Kelly, A. Mobile Robotics; Cambridge University Press: New York, NY, USA, 2013; Volume 9781107031, pp. 1–701.

29. Zhou, T.; Dickson, J.L.; Geoffrey Chase, J. Autoregressive Modeling of Drift and Random Error to Characterize a Continuous Intravascular Glucose Monitoring Sensor. J. Diabetes Sci. Technol. 2018, 12, 90–104.

30. Cleveland, R.B.; Cleveland, W.S.; McRae, J.E.; Terpenning, I. STL: A seasonal-trend decomposition. J. Off. Stat. 1990, 6, 3–73.

31. Killick, R.; Fearnhead, P.; Eckley, I.A. Optimal detection of changepoints with a linear computational cost. J. Am. Stat. Assoc. 2012, 107, 1590–1598.

32. Siegmund, D.; Venkatraman, E.S. Using the Generalized Likelihood Ratio Statistic for Sequential Detection of a Change-Point. Ann. Stat. 1995, 23, 255–271.

33. Oskiper, T.; Poor, H.V. Online activity detection in a multiuser environment using the matrix CUSUM algorithm. IEEE Trans. Inf. Theory 2002, 48, 477–493.

34. Montes De Oca, V.; Jeske, D.R.; Zhang, Q.; Rendon, C.; Marvasti, M. A cusum change-point detection algorithm for non-stationary sequences with application to data network surveillance. J. Syst. Softw. 2010, 83, 1288–1297.

35. Zhang, X.; Woodall, W.H. Dynamic probability control limits for risk-adjusted Bernoulli CUSUM charts. Stat. Med. 2015, 34, 3336–3348.

36. O' Mahony, N.; Campbell, S.; Carvalho, A.; Harapanahalli, S.; Velasco Hernandez, G.; Krpalkova, L.; Riordan, D.; Walsh, J. Deep Learning vs. Traditional Computer Vision. In Advances in Computer Vision; Chapter Deep Learn; Springer: Cham, Switzerland, 2019; pp. 128–144.

37. Han, S.W. Efficient Change Detection Methods for Bio and Healthcare Surveillance; Georgia Institute of Technology: Atlanta, GA, USA, 2010.

38. Parisi, G.I.; Kemker, R.; Part, J.L.; Kanan, C.; Wermter, S. Continual Lifelong Learning with Neural Networks: A Review. Neural Netw. 2019, 113, 54–71.

39. Finn, C.; Abbeel, P.; Levine, S. Lifelong Few-Shot Learning. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.

40. Finman, R.; Whelan, T.; Kaess, M.; Leonard, J.J. Toward lifelong object segmentation from change detection in dense RGB-D maps. In Proceedings of the 2013 European Conference on Mobile Robots (ECMR 2013), Barcelona, Spain, 25–27 September 2013; pp. 178–185.

41. Paavilainen, P.; Korhonen, I.; Lötjönen, J.; Cluitmans, L.; Jylhä, M.; Särelä, A.; Partinen, M. Circadian activity rhythm in demented and non-demented nursing-home residents measured by telemetric actigraphy. J. Sleep Res. 2005, 14, 61–68.

42. Wang, S.; Skubic, M.; Zhu, Y. Activity density map visualization and dissimilarity comparison for eldercare monitoring. IEEE Trans. Inf. Technol. Biomed. 2012, 16, 607–614.

43. Huang, R.; Sun, S. Kernel regression with sparse metric learning. J. Intell. Fuzzy Syst. 2013, 24, 775–787.

44. Weinberger, K.Q.; Tesauro, G. Metric Learning for Kernel Regression. In Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, AISTATS 2007, San Juan, Puerto Rico, 21–24 March 2007.

45. Taha, A.; Chen, Y.T.; Misu, T.; Shrivastava, A.; Davis, L. Unsupervised data uncertainty learning in visual retrieval systems. arXiv 2019, arXiv:1902.02586.

46. Shi, Y.; Bellet, A.; Sha, F. Sparse Compositional Metric Learning. arXiv 2014, arXiv:1404.4105.

47. Ying, Y.; Huang, K.; Campbell, C. Sparse Metric Learning via Smooth Optimization. In Proceedings of the 22nd International Conference on Neural Information Processing Systems (NIPS'09), Vancouver, BC, Canada, 7–10 December 2009.

48. Liu, Y.; Jun, E.; Li, Q.; Heer, J. Latent Space Cartography: Visual Analysis of Vector Space Embeddings. Comput. Graph. Forum 2019, 38, 67–78.

49. Aiordachioaie, D.; Popescu, T.D. Change Detection by Feature Extraction and Processing from Time-Frequency Images. In Proceedings of the 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI 2018), Iasi, Romania, 28–30 June 2018.

50. Hajij, M.; Zamzmi, G.; Cai, X. Persistent Homology and Graphs Representation Learning. arXiv 2021, arXiv:2102.12926.

51. Munch, E. A User's Guide to Topological Data Analysis. J. Learn. Anal. 2017, 4, 47–61.

52. Hido, S.; Idé, T.; Kashima, H.; Kubo, H.; Matsuzawa, H. Unsupervised change analysis using supervised learning. In Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin/Heidelberg, Germany, 2008; Volume 5012, pp. 148–159.

53. Aminikhanghahi, S.; Cook, D.J. A survey of methods for time series change point detection. Knowl. Inf. Syst. 2017, 51, 339–367.

54. Fallati, L.; Savini, A.; Sterlacchini, S.; Galli, P. Land use and land cover (LULC) of the Republic of the Maldives: First national map and LULC change analysis using remote-sensing data. Environ. Monit. Assess. 2017, 189, 1–15.

55. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. Int. J. Comput. Vis. 2016, 128, 336–359.

56. Shi, S.; Zhang, X.; Fan, W. A Modified Perturbed Sampling Method for Local Interpretable Model-agnostic Explanation. arXiv 2020, arXiv:2002.07434.

57. O'Mahony, N.; Campbell, S.; Carvalho, A.; Krpalkova, L.; Velasco-Hernandez, G.; Riordan, D.; Walsh, J. Understanding and Exploiting Dependent Variables with Deep Metric Learning. In Advances in Intelligent Systems and Computing; Springer: Berlin/Heidelberg, Germany, 2021; Volume 1250, pp. 97–113.

58. Zhu, S.; Yang, T.; Chen, C. Visual Explanation for Deep Metric Learning. arXiv 2019, arXiv:1909.12977.

59. Verma, S.; Dickerson, J.; Hines, K. Counterfactual Explanations for Machine Learning: A Review. arXiv 2020, arXiv:2010.10596.

60. Borghesi, A.; Baldo, F.; Milano, M. Improving Deep Learning Models via Constraint-Based Domain Knowledge: A Brief Survey. arXiv 2020, arXiv:2005.10691.

61. Seo, S.; Liu, Y. Differentiable Physics-informed Graph Networks. arXiv 2019, arXiv:1902.02950.

62. Cranmer, M.; Greydanus, S.; Hoyer, S.; Research, G.; Battaglia, P.; Spergel, D.; Ho, S. Lagrangian Neural Networks. arXiv 2020, arXiv:2003.04630.

63. Greydanus, S.; Dzamba, M.; Yosinski, J. Hamiltonian Neural Networks. arXiv 2019, arXiv:1906.01563.

64. Minervini, P.; Demeester, T.; Rocktäschel, T.; Riedel, S. Adversarial Sets for Regularising Neural Link Predictors. arXiv 2017, arXiv:1707.07596.

65. Rocktäschel, T.; Singh, S.; Riedel, S. Injecting Logical Background Knowledge into Embeddings for Relation Extraction. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp. 1119–1129.

66. Gsponer, S.; Costabello, L.; Van, C.L.; Pai, S.; Gueret, C.; Ifrim, G.; Lecue, F. Background Knowledge Injection for Interpretable Sequence Classification. arXiv 2020, arXiv:2006.14248.