Neural Computing and Applications (2023) 35:15095–15108
https://doi.org/10.1007/s00521-023-08522-z

ORIGINAL ARTICLE

Scarce data driven deep learning of drones via generalized data distribution space

Chen Li[1] · Schyler C. Sun[1] · Zhuangkun Wei[1] · Antonios Tsourdos[1] · Weisi Guo[1,2]

Received: 10 August 2022 / Accepted: 21 March 2023 / Published online: 6 April 2023
© The Author(s) 2023

Abstract

Increased drone proliferation in civilian and professional settings has created new threat vectors for airports and national infrastructures. The economic damage for a single major airport from drone incursions is estimated to be millions per day. Due to the lack of balanced representation in drone data, training accurate deep learning drone detection algorithms under scarce data is an open challenge. Existing methods largely rely on collecting diverse and comprehensive experimental drone footage data, artificially induced data augmentation, transfer and meta-learning, as well as physics-informed learning. However, these methods cannot guarantee capturing diverse drone designs and fully understanding the deep feature space of drones. Here, we show how understanding the general distribution of the drone data via a generative adversarial network (GAN), and explaining the under-learned data features using topological data analysis (TDA) can allow us to acquire under-represented data to achieve rapid and more accurate learning. We demonstrate our results on a drone image dataset, which contains both real drone images as well as simulated images from computer-aided design. When compared to random, tag-informed and expert-informed data collections (discriminator accuracy of 94.67%, 94.53% and 91.07%, respectively, after 200 epochs), our proposed GAN-TDA-informed data collection method offers a significant 4% improvement (99.42% after 200 epochs). We believe that this approach of exploiting general data distribution knowledge from neural networks can be applied to a wide range of scarce data open challenges.

Keywords Air transport · Drones · Airport safety · Discriminative neural networks · Feature distribution · Training data collection

1 Introduction

Increased proliferation of drones and autonomous air vehicles can disrupt critical national services (*e.g.*, Gatwick Airport 2018). The economic damage for air transport is estimated to be millions per day for airports and airlines [1–3]. The growth of drone industry generates high contributions to the economy (1.9% of UK GDP and supports over 600,000 jobs, 5 million consumer drone shipments worldwide in 2020 [4]), but also brings new threat factors to air transport [5, 6]. While many are amateur drones that pose no malicious intention, some may carry deadly capability and cause severe economic damage to critical infrastructure. Protection against drones is critical to ensuring smooth operation of services, while safeguarding it against the most severe threats. High-resolution cameras can classify drones using deep learning (DL), but accurate identification is critical for not disrupting normal day-to-day operations and maintaining an efficient economy [7].

Chen Li and Schyler C. Sun have equally contributed to this work.

✉ Chen Li
c.li.21@cranfield.ac.uk

Schyler C. Sun
Schyler.Sun@cranfield.ac.uk

Zhuangkun Wei
Zhuangkun.Wei@cranfield.ac.uk

Antonios Tsourdos
a.tsourdos@cranfield.ac.uk

Weisi Guo
Weisi.Guo@cranfield.ac.uk

[1] Digital Aviation Research Technology Centre (DARTeC), Cranfield University, College Road, Cranfield MK43 0AL, Bedfordshire, UK

[2] Alan Turing Institute, 96 Euston Road, London NW1 2DB, England, UK

As shown in Fig. 1, the feature of drones led to a high data scarcity, and significantly challenged the accuracy and reliability of data-driven DL drone identification. As shown in Fig. 1I, while the upper limit of accuracy for image classification has been increased by more complex deep learning architectures, the upper accuracy of DL model also limited by its logarithmic growth to the size of the training dataset [8]. This often means a large amount of resources and time is dedicated to broad data collection and (re)training DL-based system models to achieve higher drone discriminate accuracy. Simultaneously, high speed and small size of drone challenge the image capture, while multi-model and numbers of shooting angle also increase the data scarcity of drone dataset. This scarce drone dataset leads to optimization error and generalization error in DL systems (see Fig. 1II). Recent methods for scarce data learning (*e.g.,* data augmentation, meta-learning) artificially create new data based on existing ones or transfer the knowledge learned from other domains. But, these methods cannot solve generalization errors related to out-of-sample data. However, collecting extra training data could address these errors as shown in Fig. 1III. (Re)training the model on additional drone data can improve system accuracy, but is also more costly than other methods that do not require new data collection. However, a target of data collection can reduce the amount of new data to be collected, saving overall DL performance improvement costs. Accordingly, it is a key open challenge to achieve extremely high accuracy by sourcing sufficient, relevant but rare training data sets (*e.g.*, rare drone design) [9].

In this paper, we show how understanding the general distribution of the drone data via a generative adversarial network (GAN), and explaining the under-learned data features using topological data analysis (TDA) can allow us to acquire under-represented data (data instances with under-learned data features) to achieve rapid and more accurate learning. The aim is to demonstrate how to find the under-represented data for the DL model training by understanding DL learning behavior, benefiting the efficiency of improving model performance by more targeted data collection.

1.1 Related work

One of the universal challenges in deep neural network training is when there is a lack of data, the out-of-sample performance cannot be guaranteed [10]. As shown in Fig. 1II, the nature of discriminative NNs' inference is to map an high-dimensional input features into a label class using the feature distribution (FD) generalized from the training data [11] (convolutional layers extract latent features maps of input images for further discriminative work in NN). While lack of training data will result in the generalization error on out-of-sample data [12–14], data scarcity can also lead to the optimization error on known training data (see - Fig. 1a). However, collecting extra training data could address these error as shown in Fig. 1b. In general discriminative NNs, the performance increases logarithmically based on volume of training data size [8], which means the marginal cost of training data for model performance improvement boosts exponentially. Due to the diminishing returns, exhaustive or randomly searching for data is not applicable to the scenarios where data collection is expensive (*e.g.*, aerospace, military). Therefore, there is a need to create a method to identify which specific new data would be important for discriminative NNs' performance improvement based on an existing dataset, as the guidance to the new data collection work, so that to reduce data collection cost. This is the motivation for this paper.

As shown in Fig. 1III, our aim is to detect the data that is important for model training (*e.g.,* hard-to-learn drone), reduce the amount of new data required for model improvement, and achieve a faster speed and higher accuracy training of DL-based drone discriminator then other data collection methods. There are related papers addressing the aforementioned challenges in the discriminative NNs training with limited data (scarce data learning). We summarize several research achievements below and compare them in Table 1.

Data augmentation improve the training set by adding slightly modified (*e.g.,* translations, rotations and flips) copies of existing data to strengthen the invariance of NNs to the aforementioned modified data [15]. However, data augmentation is only based on raw training data, hence cannot offer additional generalization on the variation of the object itself other than its position to NNs. Transfer learning reduce the training cost of new DL model by reusing the convolutional kernels from other related well-trained DL models [16]. By doing so, NNs can partially transfer the generalization ability got in the former relevant training into the latter target inference. Similarly, meta-learning tries to abstract more universal generalization ability from multiple training domains and "learn to learn" fast [17], which lays the foundation for the few-shot learning [18]. However, the performance gain via these two methods are not guaranteed, since the transferred generalization ability is context-agnostic (*i.e.*, does not focus on the properties of the new data), which may not match the need to specific requirements. Physical-informed learning is designed to embed given laws of physics (*e.g.*, general nonlinear partial differential equations) into NNs to inform its generalization [19, 20]. Hence, there is less need for the diversity of the training data and more training data can be generated numerically from the given laws of physics. However, in most discriminative NNs applications, physical law is unknown.

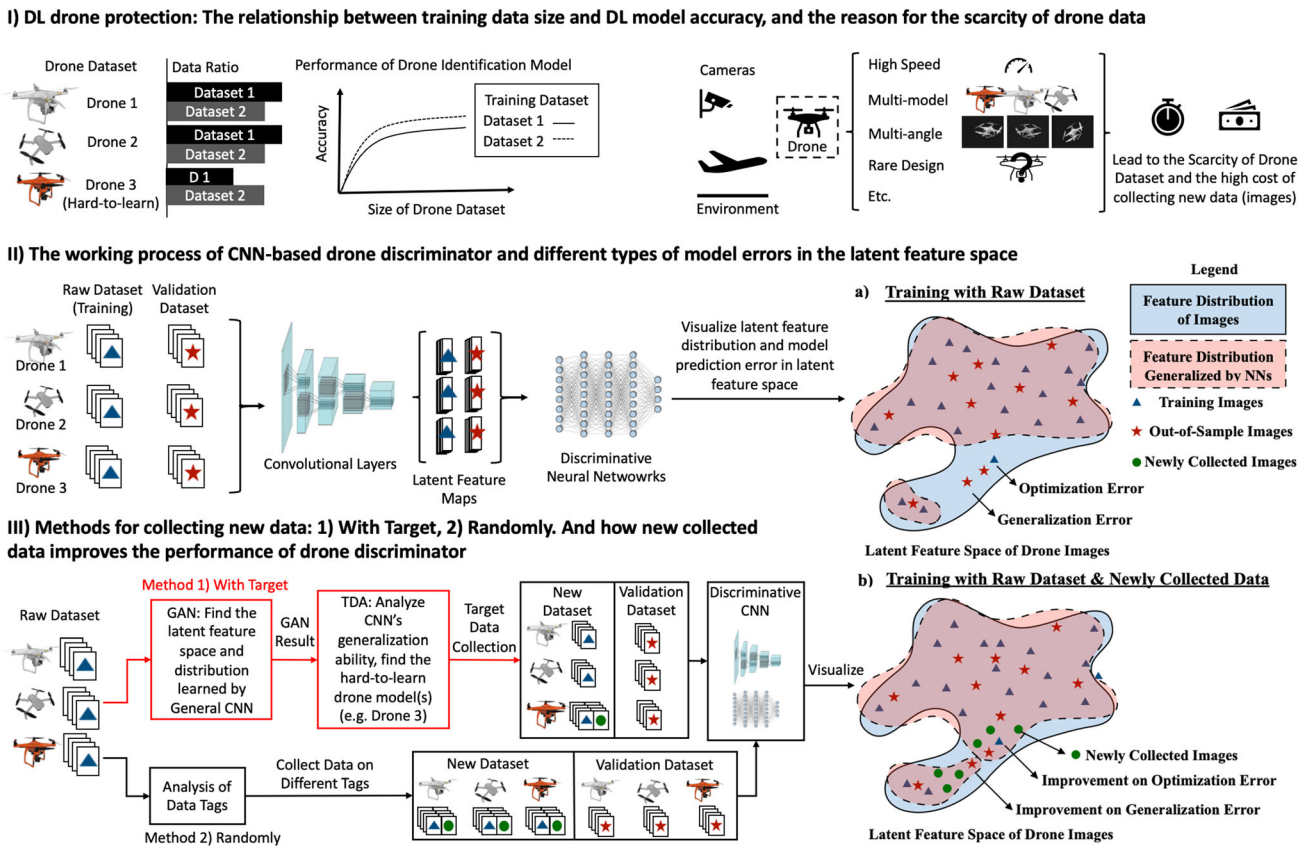

Fig. 1 Reasons for the scarcity of drone data, errors in CNN-based drone discriminators, and methods for collecting new data. **I** The upper limit of DL drone classification accuracy is affected by the training data size, but various factors of drones lead to scarcity of drone data and high acquisition costs. **II** the nature of discriminative NNs' inference is to map high-dimensional input features into a label class using the feature distribution (FD) generalized from the training data (convolutional layers extract latent features maps of input images for further discriminative work in NN), while lack of training data will result in the generalization error on out-of-sample data. **III** Our aim is to detect the data that important for model training (*e.g.,* hard-to-learn drone), reduce the amount of new data required for model improvement, and outperform randomly data collection method in both learning speed and predication accuracy of trained model

Table 1 A comparison of methods for training deep learning models with scarce data

	Data augmentation [15]	Transfer learning [16]	Meta learning [17]	Physical-informed learning [19]	Proposed GAN-TDA Method
Methodology	Add modified data into dataset	Reuse Layers from other well-trained DL models	Reuse layers trained in previous tasks	Apply physical properties in model training	Purposefully collect new data
Generalization Ability	Modified dataset	Other related data	Other related data	Data physical properties	Newly collected data
Advantage	No additional data required	Wide applicability; fast deploy speed	Learn to learn; learn fast for new tasks	Controllable generalization	Explainability; less data to be collected;
Disadvantage	Limited improvement; lack of explainability	Context-agnostic, lack of explainability	Context-agnostic, lack of explainability	Difficulties in processing physical properties	Need additional cost for collecting targeted new data

Although data augmentation enhances the training set by adding augmented data based on observed data (auxiliary variables method), there is no effect on unobserved data (*e.g.,* Drone images by un-observed angle). However, newly collected training set could address this problem. Transfer learning and meta-learning can save the training time for new tasks, but cannot enhance a trained model and are lack of explainability. Physical-informed learning

needs expert experience which is abstract and uncontrollable. Thus, we propose a method to reveal the relationship between NN's generalization ability and the composition of training data, so that to provide a feature-based target for new data collection to save the cost of collecting new data. Although GAN-TDA method is with higher complexity in calculations (needs training of new DL models) and workloads (needs collecting new data) compared with the aforementioned methods, it is still necessary for several situations: when training dataset is not representative enough, new data collection is necessary to achieve higher DL performance; and when collecting certain types of data is expensive (*e.g.,* drone data), an explanation of DL errors related to data is needed.

1.2 Innovation: GAN-TDA framework

In conventional work, methods focus on the generalization ability of the model itself, which are general methods with versatility for various applications. By contrast, the generalization ability brought by the training data attracts less attention. In this paper, we aim to identify the required data for discriminative NNs' generalization error reduction, by analyzing the existing training data through its potential feature distribution (FD) and that generalized by NNs (see Fig. 1b).

Generative adversarial network

Generative models are designed to generate data with the same FD as that learned by NNs. In principle, most common generative models, including variational autoencoder (VAE) and variations of generative adversarial network (GAN), are trained to convert the initial distribution of latent variables into that learned by NNs with training data [21–25]. For scarce data problems such as our drone detection application, one might be interested in using GAN with the following reasons: (1) GAN is proven to be asymptotically consistent in FD approximation, while VAE may have bias due to the variational lower bound, thus the generated data from GAN would be more precise in representing the FD learned by the NNs [26]; (2) With the same training data, the discriminator in GAN gives similar but more smooth convolution kernels as that in CNN [27]. More specifically, kernels in GAN have the same generalization way but weaker ability as CNN. Accordingly, the analysis in GAN can be considered as the representation of general CNN. (3) The discriminator in GAN can be considered as the pre-trained target discriminative NN for methodology validation so as to eliminate generalization ability bias caused by another arbitrary NNs [28].

In our experiment, color information in each pixel of drone images are use as inputs, the viewing of raw feature space built by pixel information will be over-dimensioned and lacks practical interpretability. Thus, to build a more informative feature space for image data, latent feature learned and processed by convolutional layers in deep learning models helps. Here, each latent feature is represented as the degree of response to a certain kernel feature in different receptive fields of the image. However, GAN does not give explicit expressions of the distribution, hence Monte Carlo synthetic data from the generator would be taken for the further FD analysis in the next step.

Topological data analysis

For high-dimensional data analysis, dimension reduction approaches, such as PCA, MDS, t-SNE and *etc*., are commonly applied [29, 30]. However, due to existing of generalization in NNs, the generated data may have more complicated topology than raw dataset in the high-dimensional feature space, while conventional methods fail to capture any structure from the data, which cause catastrophic lose in high-dimensional distance information for our analysis after dimension reduction. In our framework, topological data analysis (TDA) mapper is proposed to address this issue. With the key idea of multidimensional persistence, TDA can capture data structure and then preserve the high-dimensional distance information with simplicial complex [31–33].

In our experiment, TDA is applied to tell the difference between the potential FD of data from raw dataset and that of synthetic data from the generator in GAN. With the high-dimensional clustering in TDA, discrete nodes are used to represent the original continuous feature distribution, while the connection between nodes indicates the distance in feature space. Before we do TDA, we mix these two dataset into one with data labels attached (*i.e*., real, synthetic), so that to maintain the consistency of the topological space in TDA. Then, by analyzing the proportion of real/synthetic data in each node, one can discover the weak nodes, which lack the synthetic data, and then identify the required data by the real data tags in these nodes to guide the new data collection.

It is worth noting that in our proposed methodology, the description for required training data cannot exceed human knowledge about the data. Although GAN-TDA approach works on the feature space in NNs, the result is still interpreted using the human feature space (tags), which may not match the need of neurons. What we can do here is to tag data with our best knowledge, so that to make the data collection guidance more targeted.

Contribution and novelty

In this paper, we propose GAN-TDA to identify which specific new data would be important for discriminative NNs' performance improvement based on an existing drone dataset, as the guidance to the new data collection

work. To our best knowledge, this paper is the first to reveal the relationship between NN's generalization ability and the composition of training data, so that to improve the model performance via newly collected data.

We make three major contributions:

(i) GAN-TDA framework is proposed to guide the new data collection. Specifically, we use GAN to capture the feature distribution in inference generalized by discriminative NNs from the training data, and TDA to identify the generalization weakness on the training data.
(ii) A drone image dataset using both real drone images as well as simulated images in CAD is established for our experiment. Each image is tagged with the drone's features (*e.g.*, model, color, frame shape, camera position, etc.) as many as we can.
(iii) We demonstrate our results on a drone image dataset, which contains both real drone images as well as simulated images from computer-aided design. When compared to random, tag-guided and expert-guided data collections (discriminator accuracy of 94.67%, 94.53% and 91.07%, respectively, after 200 epochs), our proposed GAN-TDA-informed data collection method offers a significant 4% improvement (99.42% after 200 epochs).

The remainder of this paper is organized as follows. In Sect. 2, we demonstrate the working flow of our proposed GAN-TDA framework. In Sect. 3, we apply our model on drone picture data for evaluation and validation. Section 4 concludes this paper and proposes the ideas for future work.

2 Method

Given a dataset with clear properties labeled in tags that are detailed enough to guide the direction of new data collectivity (*e.g.,* images in a vehicle dataset labelled with the vehicle's maker, model, type, color, number of wheel *etc.*), our methodology is to identify which kind of data the discriminative model has weak generalization on by viewing the data distribution in the dataset and distribution learned by deep learning (DL) models.

2.1 Step 1) Learn data distribution by GAN

The first step is to find the high-dimensional distribution of the raw dataset (drone images). According to the demonstration of *Step* 1) in Fig. 2, two networks named generator G and discriminator D with different network structures will be established and initialized with individual aim to generate synthetic images and to recognize the input image is real or synthetic. Before the training of GAN, all the image from the raw dataset will firstly be pre-processed into the same size (*e.g.,* 64*64 pixels) and secondly be normalized into the same scale. These steps are to ensure each input parameter (pixel intensities in each color channel) has a similar data distribution to guarantee the convergence of DL models [34].

Processed images from the raw dataset (real image dataset) will be divided into batches $B = \{B_1, B_2 ... B_i\}$ with a fixed size (*e.g.,* 64 pictures per batch). During the training, a batch of initialized noise set n that follows a certain distribution (*e.g.,* Gaussian noise) will be generated whose batch size is the same as data batch size (*e.g.,* 64*[100 samples from Gaussian noise]). The initialized noise set will be reproduced at the beginning of each training iteration and then be processed into a set of synthetic images $G(n)$ by fractionally strided convolutions [35] in G. The optimization of G is expressed by minimizing the generative loss, which could be quantified by the discriminative result from D. And the optimization of D is to minimize the discriminative loss on both synthetic images and raw images.

During the quantification of model loss, raw images B_i will be labeled as *True* and generated synthetic data $G(n)$ will be labeled as *False*, and these information are stored into the real and synthetic tags matrix $T_{\text{tag}}(G(n), B_i)$ (mixed dataset). D will scoring the real and synthetic rate on both B_i and $G(n)$, the generated scores are stored in the real or synthetic score matrix as $T_{\text{prediction}}(G(n), B_i) = [D(G(n)), D(B_i)]$. The generative and discriminative loss could be further expressed by the divergence between $T_{\text{tag}}(G(n))$ and $T_{\text{prediction}}(G(n))$ and that between $T_{\text{tag}}(G(n), B_i)$ and $T_{\text{prediction}}(G(n), B_i)$. The divergence quantification uses Wasserstein distance to guarantee the stability of model training and avoid the collapse mode issue [24].

The processes of training a DCGAN with Wasserstein distance is shown as Algorithm 1. During the model training, optimizing model parameters by backpropagation in both G and D will let the distribution of generated synthetic data $G(n)$ gradually approaching that of the real dataset B. After the model converges, the distribution of the raw dataset is considered to have been learned and captured by GAN that the distributions of the GAN generated synthetic image set and the raw dataset are in a high similarity. Simultaneously, the ability to convert the given certain distribution noise set into real-enough synthetic data which follows the distribution of the raw dataset will be hiddenly stored with the form of model parameters in the GAN generator.

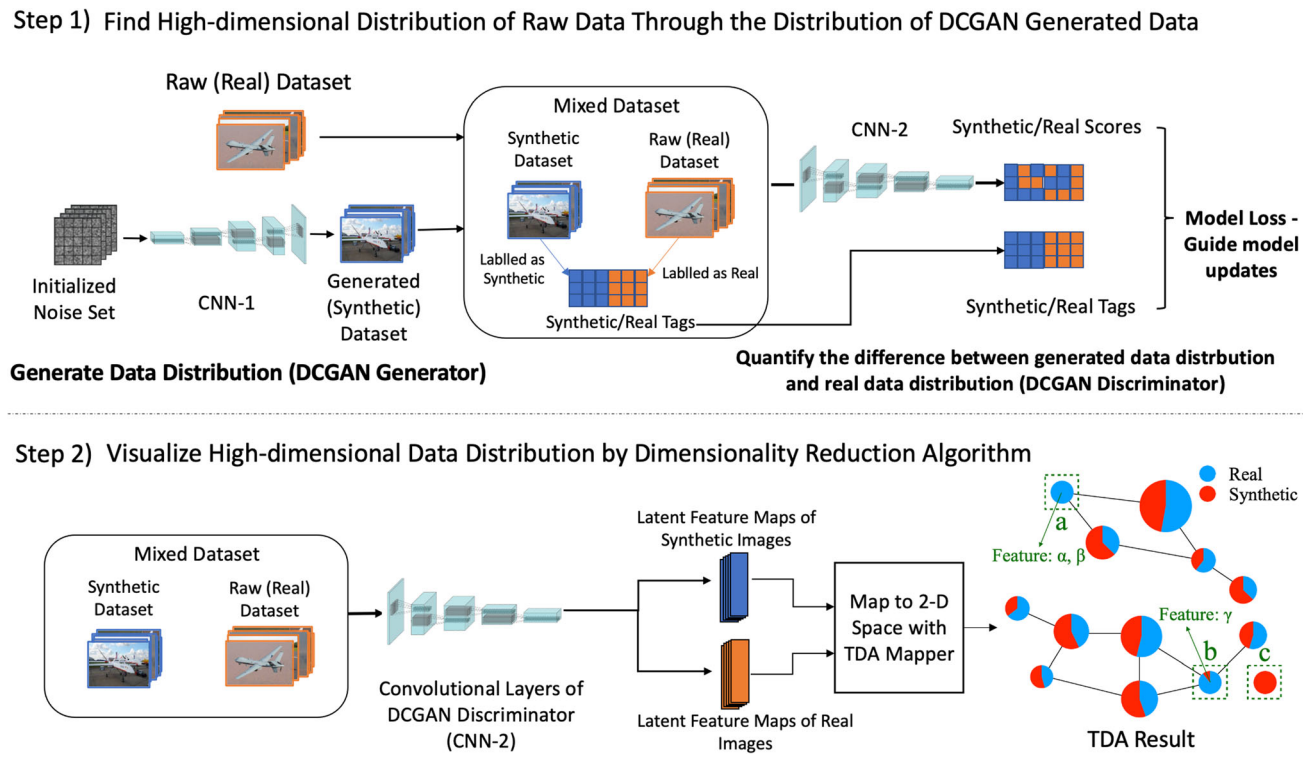

Fig. 2 Demonstration of method: Step 1) Find high-dimensional (latent features) distribution of raw dataset by DCGAN; Step 2) Use the convolutional layers of the DCGAN discriminator as the latent feature maps extractor of both the synthetic dataset and the raw dataset and use TDA to visualize their differences in data distributions

Algorithm 1 Training of DCGAN with Wasserstein distance

Require: Training set $B = \{B_1, B_2 ... B_i\}$
Require: Generator G, Discriminator D, Gaussian Noise n, Wasserstein distance calculator W, Optimizer Opt
Require: Number of total epochs j
  Initialize G, D
  for epoch $<= j$ **do**
    for $B_i \in B$ **do**
      Initialize Gaussian Noise n
      $T_{tag}(G(n), B_i) =$
        $[False * G(n), True * B_i]$
      $T_{tag}(G(n)) = [False * G(n)]$
      $T_{prediction}(G(n), B_i) =$
        $[D(G(n)), D(B_i)]$
      $T_{prediction}(G(n)) = [D(G(n))]$
      $loss\_D =$
        $W(T_{tag}(G(n), B_i), T_{prediction}(G(n), B_i))$
      $loss\_G =$
        $W(T_{tag}(G(n)), T_{prediction}(G(n)))$
      $D \leftarrow Opt(loss\_D, D)$
      $G \leftarrow Opt(loss\_G, G)$
    end for
    epoch = epoch + 1
  end forreturn D, G

2.2 Step 2) Latent feature maps extraction and TDA

In CNN, convolutional kernels are designed with weight sharing property, and each multidimensional kernel representing a unique hidden feature. So, in the convolution process, the output of each convolutional kernel means the degree of activation of an image property in a series of adjacent receptive fields controlled by stride.

The latent feature map of an image contains a set of activation degrees on different high-dimensional features and each value in the latent feature map can be seen as the activation intensity of different convolution kernels. It can also be regarded as the expression of high-dimensional image features to a low-dimensional space like the encoded latent features by generative models like VAE. The latent feature map can express the characteristics of the image to a certain extent, and be used to reduce the image dimension to facilitate the analysis of the image.

Certain data distribution will be expressed differently by latent feature spaces formed from different convolutional layers. To appropriately express the feature space for the raw dataset distribution, a proper multi-dimensional scale needs to be confirmed with a set of latent features learned by convolutional kernels. According to the fact that front convolutional layers tend to learn lower-dimensional visual

features (*e.g.,* line, Polyline, arc), later convolutional layers trying to extract high-dimensional latent features which contain complex information about positioning and relationship based on features extracted by the front layers. The feature space formed by kernels from the last convolutional layer is chosen to keep the deeply high-dimensional global information used for feedforward layers inferences. And this space could clearly express the distance between distributions of the synthetic dataset and raw dataset.

As shown in Algorithm 2, the latent feature map of a given image is defined as the output of GAN-discriminator's last convolutional layer after feeding the image into D (Step 2 of Fig. 2). Suppose the convolutional layers in D is defined as a layer set $C = \{c_1, c_2 ... c_k\}$ and the generated synthetic image from G as $S = \{G(n_1), G(n_2) ... G(n_i)\}$ (i is the batch number in B: to guarantee the generated synthetic images have the same amount of data as raw dataset). The tags of all synthetic image is set to *False* as $T_{\text{tags}}(S) = [S * False]$, and *True* for all images from real dataset B as $T_{\text{tags}}(B) = [B * True]$. As shown in Algorithm 2, the algorithm will return a dataset that contains both the latent feature maps for synthetic images L_S and real images L_B with their tags $T_{\text{tags}}(S)$ and $T_{\text{tags}}(B)$ for further TDA use.

Algorithm 2 Extraction of latent feature maps

Require: Real image dataset $B = \{B_1, B_2 ... B_i\}$, Synthetic image dataset $S = \{G(n_1), G(n_2) ... G(n_i)\}$
Require: Convolutional layers of DCGAN's discriminator $C = \{c_1, c_2 ... c_k\}$
Require: Tags for synthetic images $T_{tags}(S)$, and for real images $T_{tags}(B)$
  Define $L_S, L_B = \{\}, \{\}$
  for $i <= len(B)$ **do**
    $l_B = B_i$
    $l_S = G(n_i)$
    for $j <= len(C)$ **do**
      $l_B = c_j(l_B)$
      $l_S = c_j(l_S)$
      $j = j + 1$
    end for
    $L_S.append(l_S)$
    $L_B.append(l_B)$
    $i = i + 1$
  end forreturn $\{L_S, L_B, T_{tags}(S), T_{tags}(B)\}$

The viewing of data distribution in the chosen feature space is barricaded by its high dimensionality, where TDA makes reasonable dimensionality reduction representation to help the discovery of distance between two high-dimensional data distributions. We use Kepler mapper for TDA visualization [33].

2.3 TDA interpretation

As shown in Fig. 2 (Step 2), the TDA mapper generates a network representation of the feature space, in which each node corresponds to a set of data with similar features. The size of the node indicates the quantity of the data assigned in it, while the color is used to distinguish whether the data is real or synthetic in the node. In ideal conditions, the synthetic data would have roughly equivalent proportion in each node. Accordingly, the categories of nodes are listed as follows:

- (a) No synthetic data are generated in this node
- (b) Both synthetic data and real data are placed in this node
- (c) Synthetic data are generated anomalously outside of the real data feature distribution

While (a) indicates the GAN (representation of DNN) is failing to generalize the data with the feature α, β, (c) indicates that the GAN is still incomplete convergence, hence we will train the GAN further while (a) or (c) nodes occur. Part of (b) nodes containing few generated synthetic data with feature γ, thus we propose to emphasize our data collection procedure on data with γ.

In perfect training, the percentage of synthetic data in each node should be close, which means discriminative NNs give roughly equivalent generalization to every models. In practical, there always has bias on the generalization to different models, which can be observed from uneven percentage in each node. Low percentage indicates the lack of generalization and vice versa.

2.4 Experimental setup

Recent deep learning drone detection models make inferences mainly based on images and video information captured by surrounding cameras. These captured visual data will be processed to complex high-dimensional latent features by convolutional layers based on image properties of different receptive fields for further inference steps completed by feed-forward layers. Therefore, various appearance and shape factors of drones designed according to different working environments and purposes challenge CNN-based drone identification models in recognition precision, generalization and robustness a lot.

To investigate which design property of drone is hard for general CNN-based classifiers to learn and discriminate, an experiment is established to apply the GAN-TDA method into a collected drone image dataset (raw image dataset). This experiment aims to prove that the drone discriminator trained using additional images collected with the GAN-TDA method performs better than the model using new images collected with the randomly method.

During the experiment, the GAN-TDA model will analyze the raw dataset and generate the guidance on which kind of data should be collected additionally. Four models with the same net settings will be trained using four different newly collected datasets (see *Method 1)* and *Method 2)* in Fig. 1III) in a control experiment to evaluate the feasibility of GAN-TDA guidance. The datasets for four groups are:

Group GAN-TDA—data from the raw dataset and additional data collected under guidance from GAN-TDA (additional data for several drone models).

Group Random—data from the raw dataset and additional data collected averagely for all data categories (additional data for all drone models).

Group Tag —data from the raw dataset and additional data collected under the guidance of labels for misclassified data (additional data for several drone models).

Group Expert—data from the raw dataset and additional data collected under guidance from experts (additional data for several drone models).

During the validation, four models trained by Group GAN-TDA, Group Random, Group Tag and Group Expert datasets, respectively, will be tested on the same validation dataset which contains images for all drone models distinct from the images in training sets. According to the performance comparison between the models trained with different Groups, the superiority of GAN-TDA-guided data in improving general target CNN-based model generalization could be viewed. Details are listed as follows.

2.5 The dataset and hardware

The raw dataset contains over 4000 pictures averagely collected from 14 popular commercial drones' 3D models[1] (*e.g.*, DJI Phantom 3, Phantom 4 and Phantom 4 pro). As shown in Fig. 3, to simulate the real pictures of flying drones caught by monitors and cameras with different angles, we randomly rotate these 3D models on *x*-, *y*-, *z*-axis and take screenshots with drone center-placed and 1800*1500 pixels resolution. During the data collection, the background of each drone model is set into black without ambient light effect to remove the high-frequency information from the background. These images are store in different folders named by their drone model's name, with an additional label file that contains a unique image ID for each collected drone image with 18 different hardware and appearance characteristics of these drones (*e.g.*, shape of propellers, number of propellers, position of floor stand). The synthetic dataset is generated by DCGAN trained on the raw dataset, with the same amount of data as the raw dataset. Each image in the synthetic dataset is labeled with an index, for identification (synthetic image) and trackback purpose.

The experiment environment is split into two part: the training of DCGAN is transferred into Cloud served by Google Cloud Platform with a VM (Ubuntu 16.04) established and 1 Tesla V100 GPU (NVIDIA-SMI 450.102.04, CUDA 11.0, 16 G Memory) embedded; the evaluation experiment is processed by 8-Core Intel Core i9 (16G Memory).

2.6 DCGAN settings

To accelerate the training speed of GAN and avoid the missing of meaningful details in the appearance, each image in the real dataset is resized into a resolution of 3*64*64 (R, G, B channels, 64*64 pixels) and processed with pixel value normalization in each channel (mean = 0.5, standard deviation = 0.5). As shown in Table 2, the DCGAN-generator is designed with five fractionally strided convolution layers to generate synthetic images with the same resolution as the pre-processed real image (3*64*64). The discriminator is designed with five convolutional layers which accept 3*64*64 images as input, and the activation function of the last layer (Sigmoid in original DCGAN [35]) is removed to meet the requirements of use Wasserstein Loss [24]. During the training of DCGAN on the raw dataset, images from the dataset will be split into batches with 64 images each and shuffled before each training epoch. Gaussian noise is chosen to provide GAN-generator with original input, and in each batch training, a randomly initialized 64*100 noise set will be processed in GAN-generator into 64 synthetic images. According to the indications from [36], discriminator should be trained before generator and with more epochs than generator. The training rate is set to five that generator will be trained once after five times training of discriminator [24] with the same training rate $3e^{-4}$ using Adam optimizer. The generating quality of generator will be checked every 400 epochs training, and part of the synthetic images are sampled and shown in Fig. 3 to demonstrate the generating quality of generator after 5000 epochs training.

2.7 TDA settings

As raw TDA metrics cannot be directly visualized, the Kepler mapper is developed to reveal the topological features of the space by constructing a graph [37]. The Kepler mapper is used in this paper to aid visual exploration. TDA takes the latent feature map (4096 dimensions) of each drone image as input. Within the mapper, a customized 2D

[1] The datasets generated during and/or analyzed during the current study are available in the *figshare* repository, DOI: https://doi.org/10.6084/m9.figshare.21905094.v1.

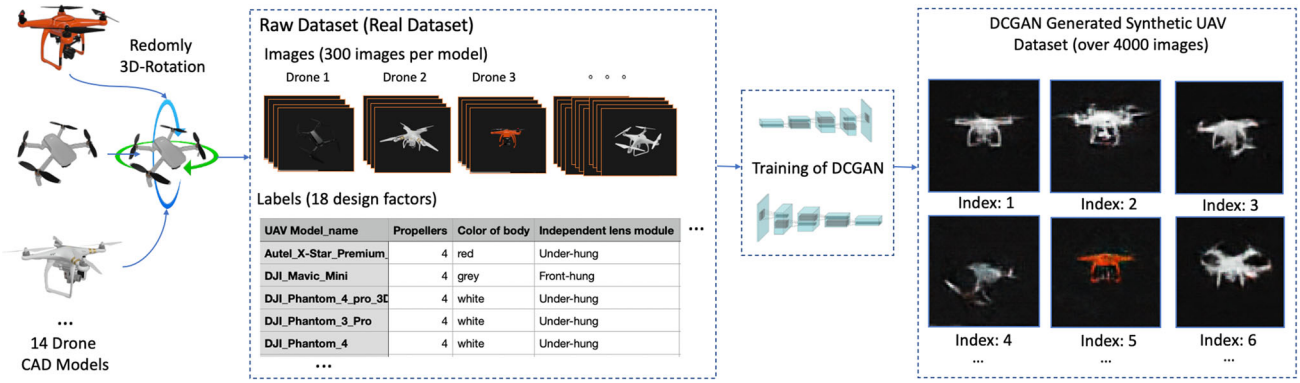

Fig. 3 Generation of the raw dataset and synthetic dataset. Raw dataset is collected from 14 CAD drone models, with 18 design factors and drone model name as labels. The synthetic dataset is generated by DCGAN trained on the raw dataset, with the same amount of data as the raw dataset

Table 2 DCGAN network training/validation settings

Generator			Discriminator		
Layer	Type	Size	Layer	Type	Size
Input	Gaussian noise	100	Input	Image	3*64*64
ConvTranspose 1	4*4 fractionally strided convolutions	512	Convolution 1	4*4 convolutions	32
-	Batch normalization	512	-	Leaky ReLU	32*32*32
-	ReLU	512*4*4	-	–	–
ConvTranspose 2	4*4 fractionally strided convolutions	256	Convolution 2	4*4 convolutions	64
-	Batch normalization	256	-	Batch normalization	64
-	ReLU	256*8*8	-	Leaky ReLU	64*16*16
ConvTranspose 3	4*4 fractionally strided convolutions	128	Convolution 3	4*4 convolutions	128
-	Batch normalization	128	-	Batch normalization	128
-	ReLU	128*16*16	-	Leaky ReLU	128*8*8
ConvTranspose 4	4*4 fractionally strided convolutions	64	Convolution 4	4*4 convolutions	256
-	Batch normalization	64	-	Batch normalization	256
-	ReLU	64*32*32	-	Leaky ReLU	256*4*4
ConvTranspose 5	4*4 fractionally strided convolutions	3	Convolution 5	4*4 convolutions	1
Output	Tanh	3*64*64	Output	None activation function/Sigmoid (validation)	1

length is established with two individual distances (Isolation Forest and L2̂-Norm) [33]. The simplicial complex is created with the customized 2D length as well as the latent feature map set, with number of intervals set to 15 and the overlap is 20%. K-means cluster is used in this paper, but any clustering algorithm could be used as advised in [37]. The number of intervals influence the number of TDA nodes, while overlap influence the overlapping among TDA nodes.

2.8 Validation settings

The evaluation of the method is to show the performances of models trained with different additional datasets on distinguishing the synthetic and real drone data.

To control the experiment variable (avoid influences brought by different initialization ways), the network in DCGAN-discriminator is chosen to be the identical network initialization whose convolutional layers are pre-trained on the raw dataset. During the training of DCGAN, the increase

in discriminative ability in discriminator is suppressed by the gradually increasing adversarial power from DCGAN-generator. Once the generator is fixed, the training of the discriminator will no longer be limited and be seen as the training of a general discriminative DNN model.

Based on the result from GAN-TDA (details are listed in the Section: TDA Result), we collect four datasets for Group GAN-TDA, Group Random, Group Tag and Group Expert, respectively. The additional data for Group GAN-TDA is evenly collected from the three detected hard-to-learn drone models (DJI Phantom 3 Pro, DJI Phantom 4 and DJI Phantom 4 pro); Group Random additional data are evenly collected from all drone models; Group Tag additional data are evenly collected from three drone models (3DR Solo, DJI Phantom 3 Pro and DJI Inspire 2) which have the highest discrimination error rate on the DCGAN discriminator (13.3%, 7.3% and 6.3%, respectively); Group Expert additional data are evenly collected from three drone models (Autel X-Star, DJI Inspire 1 and DJI Spark) which are with higher complexity in canopy structures other than others. In the practice environment, data accessibility for different models varies. Therefore, the evenly collected method is used in this experiment to represent the general random collection of new data. To control variables, we collect 100 additional images per drone model for Group GAN-TDA, Group Tag and Group Expert. Twenty-one additional images among all 14 drone models are collected for Group Random. With the use of discriminator as the initialization, the network structures of models trained by different Groups are the same as shown in Table 2, but Sigmoid activation function is added to the last feed-forward layer for the binary classification task.

The validation set is collected on all drone models evenly (25 images per drone model) and independent from the raw dataset and any additional dataset. Each image in the validation set will be pre-processed as the training sets before model inference. During training, model performance on the validation set will be supervised in each epoch.

3 Results

3.1 TDA result

The outcome by TDA shows a topological analysis result to the distribution of both synthetic data and real data. By analyzing the TDA diagram shown in Fig. 4, we found three drone models out of 14 are more difficult for deep learning models to learn. The TDA result analysis is with two steps:

Step 1—Summarize weak TDA nodes: According to the *mapper summary*, we can see none (a) or (c) type TDA node (clarified in Fig. 2) occurs that the trained GAN do capture the data distribution of the raw dataset. There are two kinds of type (b) TDA nodes: Balance TDA node (the amount of real data and synthetic data is balance) and Weak TDA node (few synthetic data in this node compared with real data). The color of nodes reflects the internal balance of data that dark color refers to weak TDA nodes with low internal data balance. The rate of real data and synthetic data in some selected TDA nodes are listed in Fig. 4 for demonstration. According to the node distribution from the mapper summary, 8.9% TDA nodes are in dark blue color so we will focus more on these weak nodes.

Step 2—Trace back to data labels: Tracing the tag of origin data whose latent feature map is placed in weak TDA nodes (which drone model are these data from) could provide clear guidance for new data collecting. The collection guidance in this paper only focused on drone model names, while guidance by other information in tags (*e.g.,* color and number of propellers) still remains for further research (due to these pieces of information in the current dataset being scarce). We list the details of an example weak TDA node as shown in Fig. 4—*Node Summary*. From here, the proportion of real data from different drone models will be summarized (*e.g.,* 16 out of 42 real data are from drone model DJI Phantom 4 Pro, and count 38.1% of real data in this node). By analyzing the data in all weak TDA nodes (with dark blue color), we found that the majority of real data placed in these nodes come from three drone models—DJI Phantom 3 Pro (22.7%), DJI Phantom 4 (27.6%) and DJI Phantom 4 Pro (33.5%). The result shows the DCGAN generalization ability on these drone models is weak. The TDA result further forms the GAN-TDA guidance that new data collection should focus on these models. During the experiment, we found that the TDA is with low sensitivity to the parameters (intervals and overlap) under current experiment settings. (During the changing of TDA parameters, DJI Phantom 4 and 4 Pro are always recognized as hard-to-learn, while sometimes DJI Mavic Pro replaces DJI Phantom 3 Pro.) However, the parameters should be adjusted for the TDA result clarity.

3.2 Discriminator result

The discriminator result shows that on the designed validation dataset, the drone detection ability of the model trained with GAN-TDA-guided additional data (Group GAN-TDA) is better than the model trained by random, tag and expert-guided additional data (Group Random, Group Tag and Group Expert).

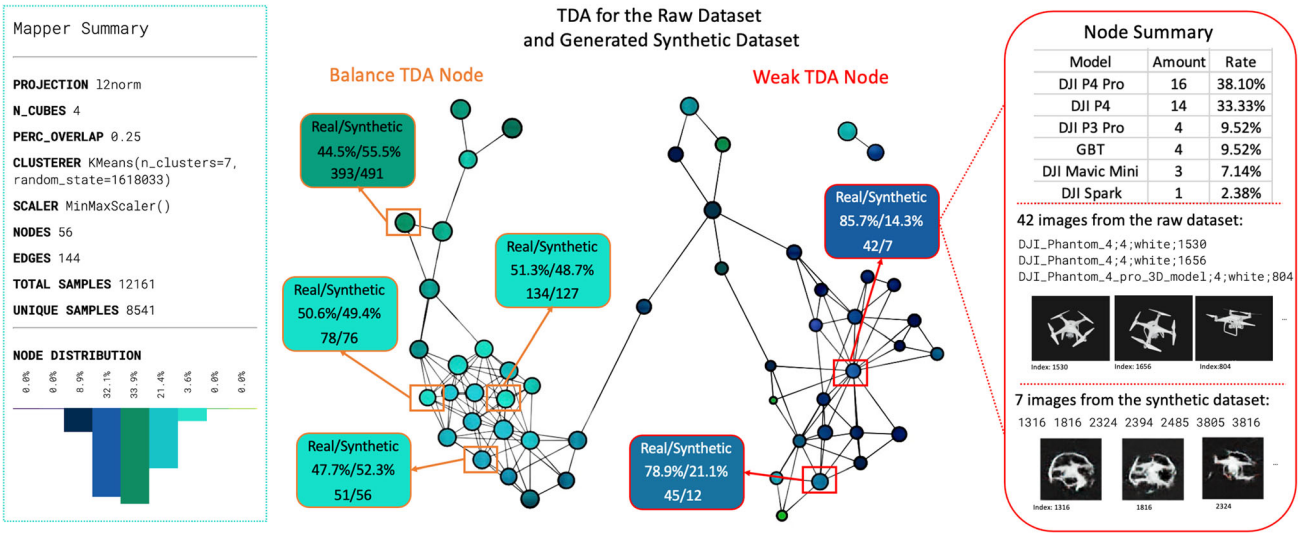

Fig. 4 TDA output and analysis

The demonstration of the discriminator result is made in two parts: *A) Validation Loss in BCE; B) Discriminative precision on the validation dataset*.

As the models' BCE losses on the validation dataset are shown in Fig. 5A, the performance of model trained by Group GAN-TDA dataset shows a significant advantage than that of models trained by Group Random, Group Tag and Group Expert dataset in BCE loss (Final loss: 0.0305, 0.1740, 0.2025 and 0.2755, respectively). This means the models' generalization ability on unseen new data could be affected by the quality of additional data for training. In other words, although different collection methods guarantee the learning of instances from the raw dataset, GAN-TDA-guided additional data performs better in boosting the model's learning of data distribution, which leads to the reinforcing of model generalization ability.

The result in *B)* demonstrates the models' discriminative precision on validation dataset. The model trained with GAN-TDA-guided additional data shows a quicker rising in inference precision of validation dataset, compared with additional data collected with other methods (Group Random, Group Tag and Group Expert). And by the end of our training, GAN-TDA-guided model's final precision on the validation dataset achieves a 4.75%/4.89%/8.35% increase in that of model trained by Group Random, Group Tag and Group Expert additional data (99.42%–94.67%/94.53%/91.07% on 350 images from verification dataset). GAN-TDA-guided additional data could make the deep learning model achieve high accuracy faster than other methods.

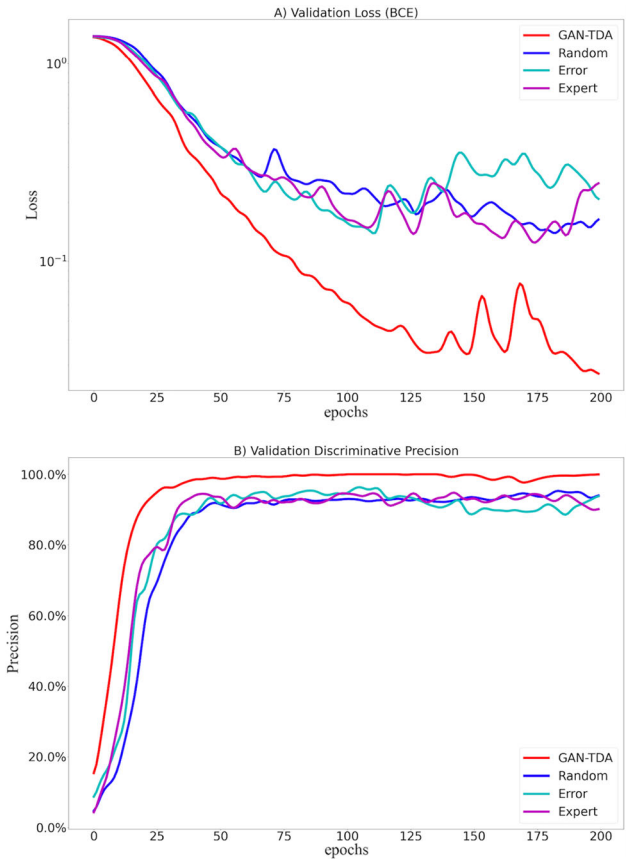

Fig. 5 Performance comparison of targeted data collection method, random data collection method, tag-informed data collection method and expert-informed data collection method

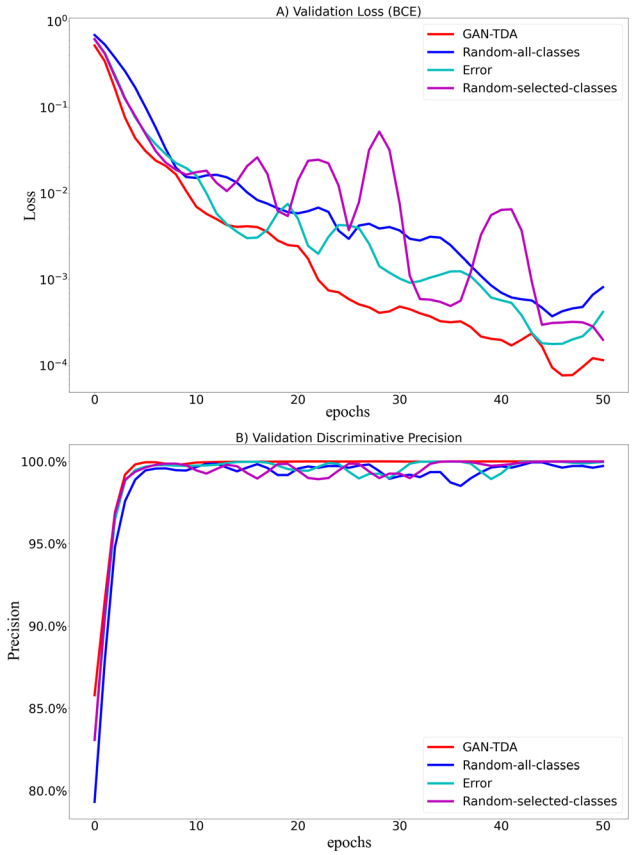

Fig. 6 Performance comparison of targeted data collection method, random data collection method (on all/selected classes) and tag-informed data collection method

4 Conclusion

High-resolution cameras using deep learning is challenged by the lack of training data sets. This often means a large amount of resources, and time is dedicated to broad data collection and (re)training the neural network—without a guaranteed convergence in improving accuracy. This paper has used explainable deep learning to identify the under-represented data and guide data collection and generation.

For high-dimensional image data, the GAN-TDA provide a solution to extract the latent features of each data instance as feature maps and generate a demonstration of the generalization ability of the convolution kernels on different latent features. With the mapping relationship among images, latent features and labels, the generalization ability of kernels on latent features could indicate that on different image properties (according to data labels). During model training, the training of hard-to-learned kernels with slow improvement in generalization abilities needs more training epochs and additional date feedings, which means an image instance with properties of these hard-to-learned kernels is more difficult for DL models to learn. (In our experiment, images with these properties are hard for GAN to generate.) Afterward, analyzing these learning-hardly images with their tagged properties can indicate the direction of new data collection. However, the GAN-TDA method is not effective, while tags are scarce—the result is not representative (*e.g.,* GAN-TDA result: two out of two canopy colors are hard-to-learn). Meanwhile, the complexity of the GAN-TDA method is much higher than other data collection methods (training the GAN model used in this paper costs over 100 hours; random data collection and expert-informed data collection do not require new models; tag-informed data collection needs to train a DL discriminator which costs much less time than GAN training). But GAN-TDA is still worthwhile when collecting new data is expensive in both time and money, or an explanation of DL errors related to data is needed.

By applying GAN-TDA proposed in our paper, we achieve a 4.75–8.35% precision boosting (99.42%) on drone discriminative NN compared with control models which use random, tag-informed and expert-informed collection methods (94.67%, 94.53% and 91.07%). Simultaneously, GAN-TDA-guided data make the discriminative NN achieve the same inference performance with less training time.

Appendix: Method verification on fashion-MNIST dataset

To verify the effectiveness of the GAN-TDA method on other datasets, we proposed a verification experiment that applies the GAN-TDA method on the open-source *Fashion-MNIST*[2] dataset. The dataset contains grayscale images (size: 28x28 pixels) of clothes and boots, with a label from 10 classes, 60k images for training and 10k images for verification. Here, we divide the validation set into two parts: the first part (0–5000 data instances) is used as a data source for collecting additional datasets, while the second part (5001–10000 data instances) still works as validation dataset.

The GAN-TDA result shows classes 1, 8, 3 and 5 are difficult to learn by convolutional layers (data rate: 35.8%, 32.3%, 9.5% and 7.4%, respectively, in weak TDA nodes). From the test result of the GAN-discriminator, the wrong discriminated data are mainly from classes 1, 6, 9 and 0 (data rate: 26.7%, 20%, 20% and 20%, respectively). Based on the information above, we design four groups of additional datasets, which are:

Group GAN-TDA—data from the raw dataset (training set) and additional data collected (from the validation set 0-5000) under classes 1, 8, 3 and 5.

[2] The Fashion-MNIST is an open-source dataset, available at: https://www.kaggle.com/datasets/zalando-research/fashionmnist.

Group Random-all-classes—data from the raw dataset (training set) and additional data collected (from the validation set 0-2000) under all classes.

Group Tag—data from the raw dataset (training set) and additional data collected (from the validation set 0-5000) under classes 1, 6, 9 and 0.

Group Random-selected-classes—data from the raw dataset (training set) and additional data collected (from the validation set 0-5000) under random 4 classes (in this paper: 4, 5, 6 and 7).

The discriminator result is shown in Fig. 6. From the figure, we found that the discriminator re-trained with GAN-TDA-guided additional data still shows a higher converge speed (see BCE loss in sub-figure *A)*). At the same time, from sub-figure *b)*, the discriminator trained with GAN-TDA-guided data reaches 100% accuracy slightly faster than that trained with additional data collected by other methods. However, the difference between GAN-TDA methods and other methods is not as large as that on the drone image dataset. The reason may be related to the ratio of the original training set to the additional data. (The amount of additional data for model training is about 7% of the original training set in the drone experiment, while that of the Fashion-MNIST experiment is only 2%.) The low ratio of additional data limits the difference in representativeness among data collected with different methods.

Funding This work is supported by the Department of Transport under the S-TRIG program 2020-21; and the EPSRC/UKRI Trustworthy Autonomous Systems Node in Security [grant number EP/V026763/1].

Data availability The datasets generated during and/or analyzed during the current study are available in the *figshare* repository, DOI: https://doi.org/10.6084/m9.figshare.21905094.v1.

Declarations

Conflict of interest Chen Li, Schyler C.Sun, Zhuangkun Wei, Antonios Tsourdos and Weisi Guo are with Digital Aviation Research Technology Centre (DARTeC), Cranfield University, Bedford, United Kingdom. Weisi Guo is also with the Alan Turing Institute, London, United Kingdom.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Vinod B (2020) The COVID-19 pandemic and airline cash flow. J Revenue Pricing Manag 19(4):228–229
2. Ball M, Barnhart C, Dresner M, Hansen M, Neels K, Odoni A, Peterson E, Sherry L, Trani A, Zou B (2010) Total delay impact study: a comprehensive assessment of the costs and impacts of flight delay in the united states. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/6234
3. Hatıpoğlu I, Tosun Ö, Tosun N (2022) Flight delay prediction based with machine learning. LogForum, 18(1)
4. Silalahi S, Ahmad T, Studiawan H (2022) Named entity recognition for drone forensic using bert and distilbert. In: 2022 international conference on data science and its applications (ICoDSA). IEEE, pp 53–58
5. Dominicus J (2021) New generation of counter UAS systems to defeat of low slow and small (LSS) air threats. In: NATO Science and Technology Organization-MP-MSG-SET-183 Specialists' meeting on drone detectability, pp KN-2-1-KN-2-20
6. Wang J, Liu Y, Song H (2021) Counter-unmanned aircraft system (s)(c-UAS): State of the art, challenges, and future trends. IEEE Aerosp Electron Syst Mag 36(3):4–29
7. Thai P, Alam S, Lilith N, Nguyen BT (2022) A computer vision framework using convolutional neural networks for airport-airside surveillance. Transp Res Part C Emerg Technol 137:103590
8. Sun C, Shrivastava A, Singh S, Gupta A (2017) Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE international conference on computer vision, pp 843–852
9. Zhu L, Yu FR, Wang Y, Ning B, Tang T (2019) Big data analytics in intelligent transportation systems: a survey. IEEE Trans Intell Transp Syst 20(1):383–398
10. Wang Y, Yao Q, Kwok JT, Ni LM (2020) Generalizing from a few examples: a survey on few-shot learning. ACM Comput Surv (CSUR) 53(3):1–34
11. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25:1097–1105
12. Jin P, Lu L, Tang Y, Karniadakis GE (2020) Quantifying the generalization error in deep learning in terms of data distribution and neural network smoothness. Neural Netw 130:85–99
13. Liu Z, Xu Y, Qiu C, Tan J (2019) A novel support vector regression algorithm incorporated with prior knowledge and error compensation for small datasets. Neural Comput Appl 31(9):4849–4864
14. Zai El Amri W, Reinhart F, Schenck W (2022) Open set task augmentation facilitates generalization of deep neural networks trained on small data sets. Neural Comput Appl 34(8):6067–6083
15. Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6(1):1–48
16. Weiss K, Khoshgoftaar TM, Wang D (2016) A survey of transfer learning. J Big data 3(1):1–40
17. Huisman M, van Rijn J. N, Plaat A (2021) A survey of deep meta-learning. Artificial Intell Rev, pp 1–59
18. Ravi S, Larochelle H (2017) Optimization as a model for few-shot learning in 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net. [Online]. Available: https://openreview.net/forum?id=rJY0-Kcll
19. Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward

and inverse problems involving nonlinear partial differential equations. J Comput Phys 378:686–707
20. Lu L, Jin P, Pang G, Zhang Z, Karniadakis GE (2021) Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. Nat Mach Intell 3(3):218–229
21. Kingma DP, Welling M (2014) Auto-encoding variational bayes. Stat 1050:1
22. Vahdat A, Kautz J (2020) Nvae: A deep hierarchical variational autoencoder. arXiv preprint arXiv:2007.03898
23. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. Adv Neural Inform Process Syst, pp 2672–2680
24. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein generative adversarial networks. In: International conference on machine learning. PMLR, pp 214–223
25. Karras T, Laine S, Aila T (2019) A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4401–4410
26. Goodfellow I (2016) Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160
27. Wang H, Wu X, Huang Z, Xing E. P (2020) High-frequency component helps explain the generalization of convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8684–8694
28. Li Y, Yosinski J, Clune J, Lipson H, Hopcroft J E (2015) Convergent learning: Do different neural networks learn the same representations?. in FE@ NIPS, pp 196–212
29. Fodor I K (2002) A survey of dimension reduction techniques. Lawrence Livermore National Lab., CA (US), Tech. Rep
30. Engel D, Hüttenberger L, Hamann B (2012) A survey of dimension reduction methods for high-dimensional data analysis and visualization. In: Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering-Proceedings of IRTG 1131 Workshop 2011. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik
31. Lum PY, Singh G, Lehman A, Ishkanov T, Vejdemo-Johansson M, Alagappan M, Carlsson J, Carlsson G (2013) Extracting insights from the shape of complex data using topology. Sci Rep 3:1236
32. Bergomi MG, Frosini P, Giorgi D, Quercioli N (2019) Towards a topological-geometrical theory of group equivariant non-expansive operators for data analysis and machine learning. Nat Mach Intell 1(9):423–433
33. Van Veen HJ, Saul N, Eargle D, Mangham SW (2019) Kepler mapper: a flexible python implementation of the mapper algorithm. J Open Sour Softw 4(42):1315
34. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. PMLR, pp 448–456
35. Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434
36. Chintala S, Denton E, Arjovsky M, Mathieu M (2016) How to train a gan? Tips and tricks to make gans work. Github. com. [Online]. Available: https://github.com/soumith/ganhacks
37. Singh G, Mémoli F, Carlsson G E, et al., (2007) Topological methods for the analysis of high dimensional data sets and 3d object recognition. PBG@ Eurographics, vol 2, pp 091–100

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.