Evaluating the Robustness of ML Models To Out-of-Distribution Data Through Similarity Analysis*

Joakim Lindén[13][0000−0002−7575−5315], Håkan Forsberg[1][0000−0002−0933−6059], Masoud Daneshtalab[1][0000−0001−6289−1521], and Ingemar Söderquist[23][0000−0001−9863−9985]

[1] Mälardalen University, Sweden
[2] Royal Institute of Technology, Sweden
[3] Saab AB, Sweden

Abstract. In Machine Learning systems, several factors impact the performance of a trained model. The most important ones include model architecture, the amount of training time, the dataset size and diversity. We present a method for analyzing datasets from a use-case scenario perspective, detecting and quantifying out-of-distribution (OOD) data on dataset level.

Our main contribution is the novel use of similarity metrics for the evaluation of the robustness of a model by introducing relative Fréchet Inception Distance (FID) and relative Kernel Inception Distance (KID) measures. These relative measures are relative to a baseline in-distribution dataset and are used to estimate how the model will perform on OOD data (i.e. estimate the model accuracy drop). We find a correlation between our proposed relative FID/relative KID measure and the drop in Average Precision (AP) accuracy on unseen data.

Keywords: datasets, neural networks, similarity metrics, accuracy estimation

1 Introduction

Properly trained models for perception tasks such as object classification, detection and semantic segmentation, show great performance in today's state-of-the-art works [17, 19]. It is however not often the case that extensive care has been put into designing the dataset used to train said models. Especially in dependable systems, the use of data driven perception functions like machine learning vision models, requires specific dataset management procedures to ensure the relevance and sufficiency of captured and/or generated data for the task [2]. To be more specific, one need to assure that the data used for training a model sufficiently spans the operating design domain (ODD) for the intended use of

* This work was partially funded by Sweden's Innovation Agency and the Swedish Foundation for Strategic Research.

**Preprint submitted to the 1$^{\text{st}}$ Workshop on Advanced AI Techniques for Data Management and Analytics (AIDMA).
To be published by Springer in the Communications in Computer and Information Science (CCIS) book series.**

the function when operating in its real-world environment. This includes scenario diversity where different scene parameters having a visual impact on the rendered scene are varied; parameters like daylight conditions, weather, location etc are examples of such scene-altering parameters.

The intention of this paper is to address the perception accuracy problem from a data OOD point-of-view. To have maximum control in our experiments, we exclude real-world captured data from our scope, and focus purely on the simulated environment. It is however a long-term objective to extend this work to incorporate captured data into the data curating process, creating a hybrid data approach that addresses also the inherent domain shift from synthetic to real-world captured data. The notion of dataset distance measures, along with a definition of a baseline in-distribution dataset, allows us to quantify a dataset's distance from the baseline dataset, and relate this to the internal variation of the baseline dataset. This makes it possible to explore different dimensions of the image space for the ODD and construct a well-balanced training set.

In this paper we direct our focus to an aviation use-case - visually detecting runways during approach - which could serve as a natural extension of the pilots' perception helping to reduce some of the workload present during critical stages of approach. The performance of our detection model is quantified by the MS COCO [8] evaluation metrics commonly used for object detection. With the ability to estimate the performance of our model in a certain part of our ODD it is possible to design our dataset to be (more) complete from the start and hence shorten accumulated model training time due to dataset updates. This will likely lead to a more controlled and efficient data management and model development phase. It is however not the primary focus of this paper to point to how performance drop estimates translate into requirements on data sampling, nor do we try to assess total completeness of datasets.

This paper is organized as follows: In section 2 we present relevant related work. In section 3 we explain our method for creating the baseline dataset and variations thereof. We also present the relative distance measures and how they're used in this context. In section 4 we present the results from our experiments, including the sampling of new positional coordinates at different locations, the effects of parameter variations to dataset distances and further data visualizations for context. We also present the correlation between accuracy scores and similarity measures. In section 5 we discuss the results and how to interpret the findings. Finally, Section 6 concludes this paper.

2 Related work

Sun et al. [12] show that the amount of data trained upon increases the accuracy of the model on a logarithmic scale. Gaidon et al. [4] successfully train models on the virtual KITTI dataset, suggesting that the use of synthetically produced data indeed can deliver performance in the real-world scenario, given that the domain gap is sufficiently small. By training on synthetic data followed by real-world data fine-tuning, they find good performance in their automotive experiments.

Freemont et al. [3] propose the Scenic probabilistic programming language and describe the use of this to find corner cases in synthetically generated scenarios in general, and the automotive domain in particular. Scenic allows the user to programmatically construct a parameterized scenario where position and orientation of objects and their inter-relations can be controlled and views of the scenario can be sampled with these variations included in the process.

Yang et al. [18] discuss out-of-distribution detection methods in general, of which some are categorized as distance-based. An example is that of Masana et al. [10] where they propose to use metric learning for anomaly and novelty detection, or Techapanurak et al. [15] where they use scaled Cosine Similarity for a hyper parameter-free ood detection. Sun et al. [13] also explore non-parametric nearest-neighbor distance for OOD detection. Zilly et al. [20] investigate the correlation between Fréchet Distance (FD) and model accuracy for two different classification tasks, finding the performance to correlate to the distance between training and test sets. Guillory et al. [5] claim FD and Maximum Mean Discrepancy (MMD) distance measures do not reliably predict performance drop due to distribution shift in natural image content. Instead, they advocate the use of Average Confidence (AC) for this purpose.

Theis et al. [16] discuss different aspects of several different measures of similarity (in their context for evaluation of Generative Adversarial Networks (GANs)) including FID and KID, both of which are used in our study.

3 Method

The first step of our method regards creating a baseline scenario. We consider this to be our unperturbed scenario and we use it to define our in-distribution dataset. If it is possible having a real-world data source to guide the parameter choices of the baseline scenario, it greatly helps this step. In our use-case we use the OpenSky [11] Automatic Dependent Surveillance–Broadcast (ADS-B) data source to establish a funnel of coordinates for a normal approach to an airport. This part is detailed in section 3.1. When the baseline scenario is defined we need to create the corresponding dataset by sampling visual representations of the scene from the simulator, in our case we use Xplane 11.

The second step in this method is to impose variations in the baseline scenario and create additional datasets for these augmented scenarios. The details of this step are layed out in section 3.2.

The third step is to measure a distance between the baseline dataset and the augmented ones, thus quantifying the degree of out-of-distribution. The details of this part of the procedure are shown in section 3.3.

Having a notion of distance between our baseline and augmented datasets, this metric may be used to estimate the expected drop in accuracy of our baseline model (i.e. trained on un-augmented baseline dataset) when exposed to the augmentations. We hypothesize that this accuracy drop estimation can translate into requirements on the amount of data augmentation needed to diminish the accuracy gap for the out-of-distribution test.

3.1 Sampling Methods

Data filtering The OpenSky historical database hosts lots of ADS-B information on air traffic movement, which we utilize to help define our in-distribution dataset. The ADS-B data is filtered to remove unwanted aircraft (operating in other modes) and we end up with a usable set of ADS-B points representing our baseline use-case in terms of aircraft positions.

In previous research [9] we go into more details on how we analyze this ADS-B dataset, using three different methods to draw new samples of aircraft position. In this paper however, we focus on one of the methods which is described below. This method works well to draw in-distribution samples and facilitates 'teleportation' of our scenario to a different geographic location without the need for ADS-B data for that particular site.

We divide the airspace into 8 bins, one nautical mile (NM) sized based on distance to runway. For each bin k, we calculate the average lateral position and standard deviation (lateral in this context is relative to runway extended centerline) ${\mu_k}^{Lat}, {\sigma_k}^{Lat}$ along with the relative altitude and standard deviation ${\mu_k}^{Alt}, {\sigma_k}^{Alt}$.

When we draw new samples we randomize distance to runway to be uniform within bin k. We find our appropriate altitude and lateral displacement by sampling the normal distributions $N({\mu_k}^{Lat}, {\sigma_k}^{Lat})$ and $N({\mu_k}^{Alt}, {\sigma_k}^{Alt})$ respectively.

3.2 Environment parameter variations

In the previous section we established a way of parameterizing our relative position to the object of interest, the runway. In this section we describe how to control other variations of our scene rendering which are not directly tied to aircraft position.

There are some limitations to the ADS-B data source. Specifically it does not include aircraft attitude information (i.e. roll, pitch and heading angles) so these still need to be estimated. Position and attitude is used to place a virtual camera with a cockpit-like view of the approaching runway - the object we are trying to detect. In our work we assert uniform variations of the attitude angles within limits of normal aircraft operations: roll angle limited to ±10 degrees, pitch angle ±3 degrees (except the closest set of images, where the camera was tilted down an extra 15 degrees) and heading was within ±3 degrees of runway heading. The variations imposed by positional and attitude variations are all included in our baseline case, i.e. they are considered in-distribution. We use Scenic to help randomize aircraft attitude in the sampling process.

We are now ready for introducing variations into our environment. For this study we have considered variations in weather (clear vs cloudy) and daylight conditions (mid day vs dusk or evening). The reasons for these choices are that we expect variations due to the effect of clouds to be quite small, whereas the variations of daylight conditions are expected to show a greater dissimilarity to the baseline case. Finally, one more experiment with dusk conditions was included to cover a moderate variation case. It is desired to have variations spanning a wider range here to get a more general understanding of how these variations impact accuracy performance in later experiments.

We sample data from our simulated environment for our different cases (baseline and augmentations). For each case we sample 1512 images, split into 1000 for training and 512 for evaluation (see Figure 1):

A_k Baseline scenario, clear weather, daylight conditions.
B_k Augmented scenario, cloudy weather, daylight conditions (approach lights sometimes on).
C_k Augmented scenario, clear weather, dark night conditions (approach lights always on).
D_k Augmented scenario, cloudy weather, dark night conditions (approach lights always on).

Finally we repeat this experiment for k different runway sites, such that we define a baseline case for each runway site and perform the same augmentation experiments. We include 4 different sites in the study.

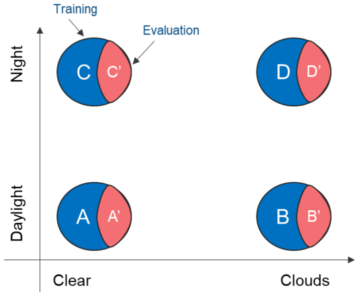

Fig. 1. Augmented and baseline datasets naming and organization. A-D are the named training datasets (represented by the blue parts of the diagram), whereas the sets A'-D' (represented by the red parts of the diagram) are used for evaluation and distance measurements. Note that the evaluation data is never used for training. The training data in set A are sampled from the same scenario and distribution as the data in A', so we can guarantee that A' is in-distribution of A; the same applies to B and B' etc.

3.3 Similarity measures

Different measures of distance exist and it is important to understand how and when to use a specific distance or similarity measure. Kullback–Leibler (KL) divergence [7] is a measure of distribution similarity defined as

$$KL(P||Q) = \sum_x P(x) \log(\frac{P(x)}{Q(x)}). \tag{1}$$

By evaluating this measure over distributions P and Q related to two different datasets D_P and D_Q we may quantify how well they align, and in a sense, whether one can be used in place of the other. t-distributed stochastic neighbor embedding (t-SNE) is a dimensionality reduction method which minimizes the KL divergence between two distributions of data point embeddings (one in high-dimensional space and the other in the reduced dimensional space). Examples of this are shown in section 4.

A more common way in the field of machine learning to measure distances between image datasets is the Fréchet Inception Distance (FID) [6] which is based on a metric distance function (the Wasserstein distance). This is based on the assumption that the feature vector representation of the different datasets is normally distributed, i.e. for two multivariate Gaussian distributed variables $X_1 \sim N(\mu_1, \Sigma_1)$ and $X_2 \sim N(\mu_2, \Sigma_2)$ the squared distance is calculated as

$$d^2 = ||\mu_1 - \mu_2||^2 + Tr(\Sigma_1 + \Sigma_2 - 2 * \sqrt{\Sigma_1 * \Sigma_2}), \tag{2}$$

where d is the distance, μ_k is the mean vector and Σ_k is the covariance matrix of the multivariate variable X_k. The vectors X_k are taken as the output of an intermediate layer of the Inception-v3 network [14], a 2048-dimensional vector.

Kernel Inception Distance [1] (KID) is another way to measure dataset similarity, which is based on the Maximum Mean Discrepancy (MMD). MMD is a distance on the space of probability measures. The distance is defined based on the notion of embedding probabilities in a Reproducing Kernel Hilbert Space (RKHS). The Hilbert space properties conveniently lends us a way of measuring distance, e.g. by the norm induced from the inner product. Let P be a probability measure on X_1 and Q be the same for X_2. Then

$$MMD^2(P, Q) = E_P[k(X_1, X_1)] - 2E_{P,Q}[k(X_1, X_2)] + E_Q[k(X_2, X_2)], \tag{3}$$

using a kernel function k. In our experiments we use a polynomial kernel function $k(x, y) = (\gamma x^\top y + c_0)^d$, with $d = 3$, $\gamma = 1/2048$ and $c_0 = 1$. Similar to FID, Inception-v3 intermediate layer outputs are used for the probability measure, however the Gaussian distribution assumption of X_k can be relaxed here.

Relative FID and KID We now have the tools for quantifying similarity. In this work we will use the FID and KID distances for our measurements. We define the relative FID (RFID) measure by the following reasoning: We measure the FID distance between our sets A and A', which should be small since we have drawn them from the same scenario and distribution. We let $K_{FID} = 1/FID(A, A')$ be a normalizing constant and then define

$$RFID(A, B) = K_{FID} * FID(A, B), \tag{4}$$

i.e. we normalize the FID score based on what is the expected distance for in-distribution data. Note that by definition $RFID(A, A') = 1$. Analogously we define $K_{KID} = 1/KID(A, A')$ and

$$RKID(A, B) = K_{KID} * KID(A, B). \tag{5}$$

4 Results

In this section we show our experimental results. The results of the following sub-sections will be analyzed in the discussion section following the results.

4.1 Sampling Method

The proposed sampling method is used to create new coordinates for previously unseen aircraft positions. Figure 2 shows how our samples are distributed laterally. We also show the possibility of this sampling method to sample at new locations, where ADS-B data might be limited. This image is published in our previous work [9].

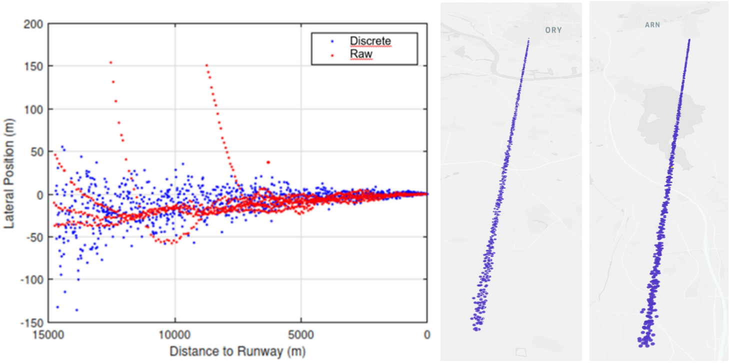

Fig. 2. Left: Data points from filtered ADS-B points (red) and generated samples (blue). Center: Generated samples at Paris-Orly. Right: Generated samples at Stockholm Arlanda. These sample points are used to set the viewing point for an aircraft approaching a runway. A visual image is generated at each view-point, rendered in Xplane flight simulator. Best viewed in colour.

4.2 Environment parameter variations

In Figure 3 we show Average Precision (AP) results from different training scenarios like training only on clear daylight data (A) and evaluating on different datasets (in- and out-of-distribution). We also show the recovery in accuracy when including some of the B, C and D data into training, which is expected since B', C' and D' are then no longer out-of-distribution.

4.3 Similarity measures

For each airport site k we have trained a model (Faster R-CNN with ResNet-50-FPN-3x backbone pretrained on ImageNet) on the baseline dataset (A_k) for that

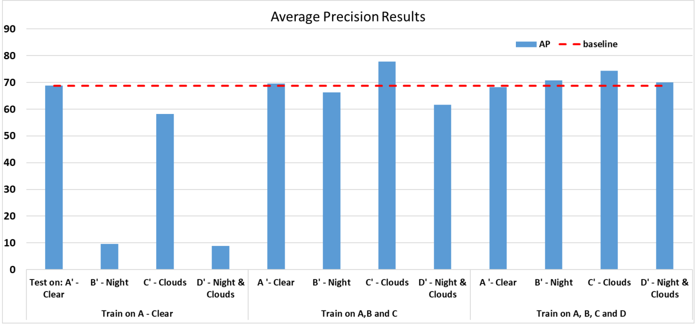

Fig. 3. Dashed line shows the baseline scenario performance. Blue bars indicate AP score on test sets (A'-D'). Accuracy drop is evident when tested on B', C' and D' OOD data. When our model is trained on data from sets A, B and C we see a clear recovery, which increase further if we add also data from set D.

site. This model was then evaluated on A'_k, B'_k, C'_k and D'_k datasets. We can thus evaluate the absolute and relative performance drop of the model. We also calculate the $RFID(A_k, B'_k)$, $RFID(A_k, C'_k)$ and $RFID(A_k, D'_k)$ and similarly the same combinations for $RKID$ measures, as shown in table 1. AP small, medium and large refers to the MS COCO [8] evaluation metrics commonly used for object detection. Here AP small only includes accuracy for objects smaller than 32^2 pixels, AP medium includes object sizes from 32^2 to 96^2 pixels and AP large includes all those objects larger than 96^2 pixels.

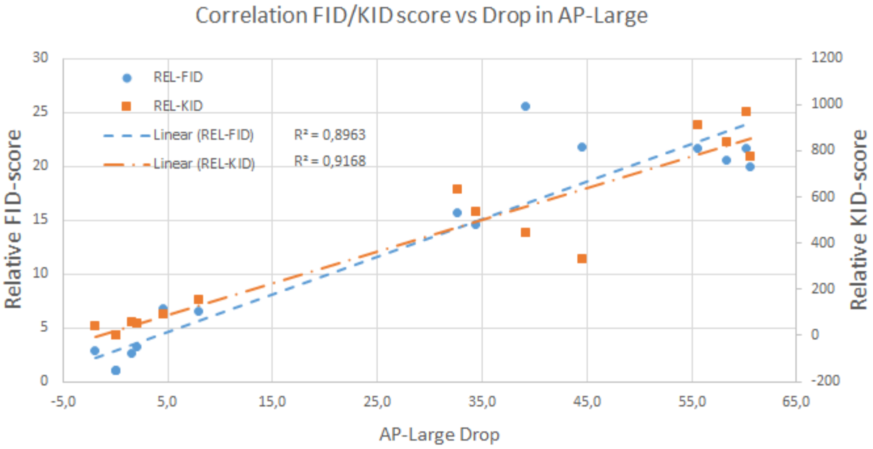

Fig. 4. Correlation between drop in AP for large objects and Relative FID/KID measures. Pearson correlation with AP-Large Drop is 0.93 for RFID and 0.94 for RKID.

Table 1. Accuracy results and corresponding RFID and KFID measurements.

Trained on	**Evaluated on**	**AP**	**AP small**	**AP medium**	**AP Large**	**RFID**	**RKID**
A_1 - Arlanda	A'_1 - Clear	68,8	59,2	86,0	90,2	1,0	1,0
	B'_1 - Night	9,6	4,1	10,2	55,8	14,6	539,1
	C'_1 - Clouds	58,2	49,2	75,1	92,1	2,9	39,5
	D'_1 - Night + Clouds	8,8	4,0	12,8	57,6	15,7	635,0
	E'_1 - Dusk	34,2	26,7	42,5	82,3	6,6	158,7
A_2 - Doha	A'_2 - Clear	83,2	75,5	89,4	90,4	1,0	1,0
	B'_2 - Night	36,6	29,0	41,2	45,8	21,7	333,0
	C'_2 - Clouds	75,1	60,4	84,8	85,8	6,7	92,3
	D'_2 - Night + Clouds	36,6	29,3	38,3	51,3	25,5	448,5
A_3 - Paris-Orly (rwy 07)	A'_3 - Clear	79,5	70,4	93,0	94,9	1,0	1,0
	B'_3 - Night	29,4	28,4	31,8	34,3	19,9	775,5
	C'_3 - Clouds	64,5	54,3	75,1	92,8	3,3	56,0
	D'_3 - Night + Clouds	28,0	26,7	29,2	39,4	21,6	913,8
A_4 - Paris-Orly (rwy 25)	A'_4 - Clear	77,5	69,2	90,6	89,8	1,0	1,0
	B'_4 - Night	34,5	32,2	48,2	31,6	20,6	841,6
	C'_4 - Clouds	74,6	67,9	84,6	88,3	2,6	60,3
	D'_4 - Night + Clouds	33,7	29,8	50,6	29,6	21,7	971,8

In Figure 4 we show the correlation between our relative dataset similarity measures (RFID, RKID) and the drop in AP accuracy for large objects. The linear regression lines included show the general correlation here. In Figure 5 we show the corresponding results for AP score across all object sizes. The correlation is less pronounced in this case. In Figure 6 we show the t-distributed stochastic neighbor embedding of our datasets. t-SNE is a way of visualizing high-dimensional data by embedding it in a lower dimensional space. In our case we use this statistical method to visually relate all the images in all our datasets.

5 Discussion

The sampling method used for generating in-distribution aircraft positions enabled us to teleport our data to other locations in the simulated world, which opened up the possibility of generating more diverse datasets, This was used to repeat our experiments at 4 different runways.

The results from our RFID and RKID metrics are quite well aligned, as both are showing a linear correlation with AP accuracy drop, though RFID was showing a slight edge over RKID when looking at the most general AP accuracy score. The RFID distance assume our feature embeddings to be normally distributed, but even though this does not seem to be the case the result shows surprising alignment with the RKID method which does not require this normality assumption. In general our results indicate a higher correlation between the similarity metrics and accuracy when looking at larger (i.e. closer) objects. The reason for

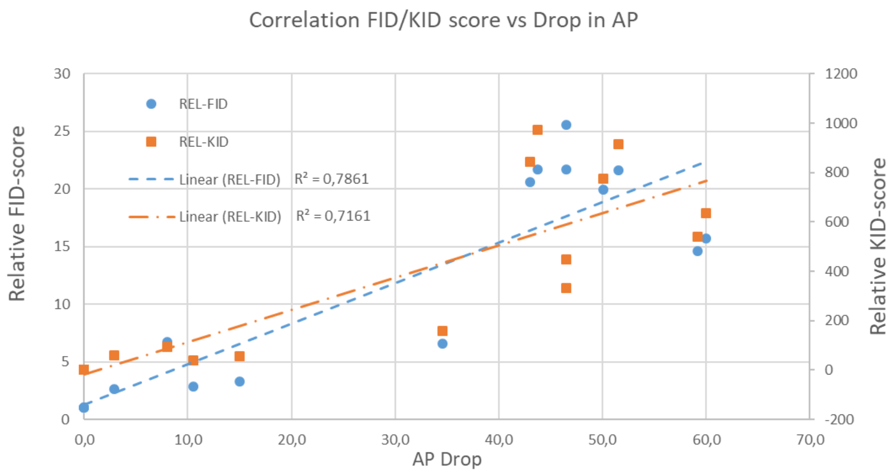

Fig. 5. Correlation between drop in AP and Relative FID/KID measures. Pearson correlation with AP Drop is 0.81 for RFID and 0.77 for RKID.

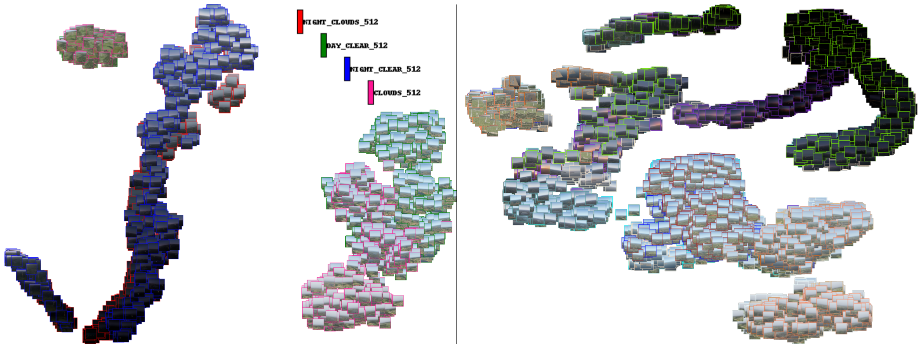

Fig. 6. T-SNE embeddings of images in datasets A'_1-D'_1 (left plot) and all datasets A'_1-D'_4 (right plot). Each colour of the image frame indicates a unique dataset. We can see some large clustering where clear and cloudy datasets gets lumped together. In the right plot we see a high degree of mixing between different sets, e.g. the day clear datasets for the different sites overlap substantially, but the night and day sets are quite distinct (as can be seen by regarding the brightness of the images). Best viewed in colour.

this is likely due to imprecise ground truth-boxes for the more distant objects, where a few pixels offset can give a large reduction in accuracy.

We note that the drop in accuracy was recovered when the training dataset was expanded to include the cloudy and night time sets (B, C and D).

The t-SNE embedding is also in general agreement with the RFID and RKID results. For instance we can see visually in Figure 6 that night (B) and night with clouds (D) are not disjoint enough to be separate categories. The t-SNE visualization is a great tool for getting a holistic view on how to parametrize the operating design domain.

6 Conclusions

We have proposed a method for analyzing the data generation process for synthetically produced visual aviation data. Specifically we introduce a way of quantifying dataset distance to expected drop in accuracy for object detection on out-of-distribution data. The correlation between RFID/RKID and drop in AP was not as clear as expected, but correlation exists. If we limit our study to the larger objects, the correlation is stronger.

We have not been able yet to address the hypothesis that the accuracy drop estimation can translate into requirements on the amount of data augmentation needed to diminish the accuracy gap for the out-of-distribution data. For this more work is needed. It should also be noted that this method should be validated in future work with other models and use-cases to see the extent to which these results generalize. Finally, since we have not looked at natural image content, we cannot make certain statements on the validity of these results in a natural image context.

Acknowledgements The authors would like to thank members of the HERO group at Mälardalen University for constructive discussions and feedback, especially from Seyedhamidreza Mousavi.

References

1. Bińkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying mmd gans. arXiv preprint arXiv:1801.01401 (2018)
2. Cluzeau, J.M., Henriquel, X., Rebender, G., Soudain, G., van Dijk, L., Gronskiy, A., Haber, D., Perret-Gentil, C., Polak, R.: Concepts of design assurance for neural networks (codann). Public Report Extract Version **1**, 1–104 (2020)
3. Fremont, D.J., Dreossi, T., Ghosh, S., Yue, X., Sangiovanni-Vincentelli, A.L., Seshia, S.A.: Scenic: a language for scenario specification and scene generation. In: Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation. pp. 63–78 (2019)
4. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtual worlds as proxy for multi-object tracking analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4340–4349 (2016)

5. Guillory, D., Shankar, V., Ebrahimi, S., Darrell, T., Schmidt, L.: Predicting with confidence on unseen distributions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1134–1144 (2021)
6. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
7. Kullback, S., Leibler, R.A.: On information and sufficiency. The annals of mathematical statistics **22**(1), 79–86 (1951)
8. Lin, T., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Doll'a r, P., Zitnick, C.L.: Microsoft COCO: common objects in context. CoRR **abs/1405.0312** (2014), http://arxiv.org/abs/1405.0312
9. Lindén, J., Forsberg, H., Haddad, J., Tagebrand, E., Cedernaes, E., Ek, E.G., Daneshtalab, M.: Curating datasets for visual runway detection. In: 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC). pp. 1–9. IEEE (2021)
10. Masana, M., Ruiz, I., Serrat, J., van de Weijer, J., Lopez, A.M.: Metric learning for novelty and anomaly detection. arXiv preprint arXiv:1808.05492 (2018)
11. Schäfer, M., Strohmeier, M., Lenders, V., Martinovic, I., Wilhelm, M.: Bringing up opensky: A large-scale ads-b sensor network for research. In: IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks. pp. 83–94. IEEE (2014)
12. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE international conference on computer vision. pp. 843–852 (2017)
13. Sun, Y., Ming, Y., Zhu, X., Li, Y.: Out-of-distribution detection with deep nearest neighbors. In: International Conference on Machine Learning. pp. 20827–20840. PMLR (2022)
14. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
15. Techapanurak, E., Suganuma, M., Okatani, T.: Hyperparameter-free out-of-distribution detection using softmax of scaled cosine similarity. arXiv preprint arXiv:1905.10628 (2019)
16. Theis, L., Oord, A.v.d., Bethge, M.: A note on the evaluation of generative models. arXiv preprint arXiv:1511.01844 (2015)
17. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696 (2022)
18. Yang, J., Zhou, K., Li, Y., Liu, Z.: Generalized out-of-distribution detection: A survey. arXiv preprint arXiv:2110.11334 (2021)
19. Zaidi, S.S.A., Ansari, M.S., Aslam, A., Kanwal, N., Asghar, M., Lee, B.: A survey of modern deep learning based object detection models. Digital Signal Processing p. 103514 (2022)
20. Zilly, J., Zilly, H., Richter, O., Wattenhofer, R., Censi, A., Frazzoli, E.: The frechet distance of training and test distribution predicts the generalization gap (2019)