Strategic Incorporation of Synthetic Data for Performance Enhancement in Deep Learning A Case Study on Object Tracking Tasks

Jatin Katyal[1] and Charalambos Poullis[1]

Immersive and Creative Technologies Lab
Concordia University, Montreal QC, Canada

Abstract. Obtaining training data for machine learning models can be challenging. Capturing or gathering the data, followed by its manual labelling, is an expensive and time-consuming process. In cases where there are no publicly accessible datasets, this can significantly hinder progress. In this paper, we analyze the similarity between synthetic and real data. While focusing on an object tracking task, we investigate the quantitative improvement influenced by the concentration of the synthetic data and the variation in the distribution of training samples induced by it. Through examination of three well-known benchmarks, we reveal guidelines that lead to performance gain. We quantify the minimum variation required and demonstrate its efficacy on prominent object-tracking neural network architecture.

Keywords: Synthetic Data · Deep Learning · Multiple Object Tracking.

1 Introduction

The process of data collecting is not without challenges, requiring substantial investments of time and resources to obtain labelled samples of high quality. Given the resource and capital costs associated with data acquisition, it is more pragmatic and cost-effective to utilize publicly available datasets that have already been published. In practice, however, straightforward application of such datasets may not always be feasible or effective due to a variety of potential challenges or constraints such as biases. One potential solution to these challenges is utilizing synthetic data as a supplement to real-world datasets. By supplementing real-world data with synthetic data, researchers can overcome the limitations inherent to traditional data sources [1, 2], thereby enhancing the overall quality and utility of their datasets.

The utilization of synthetic data has been extensively documented in recent literature, as evidenced by multiple works [33, 38, 30]. However, much of this prior research has focused on domain adaptation techniques which is adding another computationally expensive step to an already resource-hungry deep learning task. This paper examines the impact of the direct use of synthetic data on the performance of machine learning models. In a preliminary step, we analyze the Frechet Inception Distance (FID) [12] between the synthetic and the real sequences for three benchmark datasets. Building upon the patterns observed, we form clusters for both low and high

The approach we propose is both simple and straightforward, involving the direct utilization of synthetic data without the need for additional domain adaptation steps during training. We justify this approach by viewing the domain adaptation step as a potentially costly and extra procedure when dealing with an already challenging task.

Our proposed strategy aligns with prior studies that have incorporated synthetic data into their training procedures without domain adaptation. However, our approach differs significantly from those

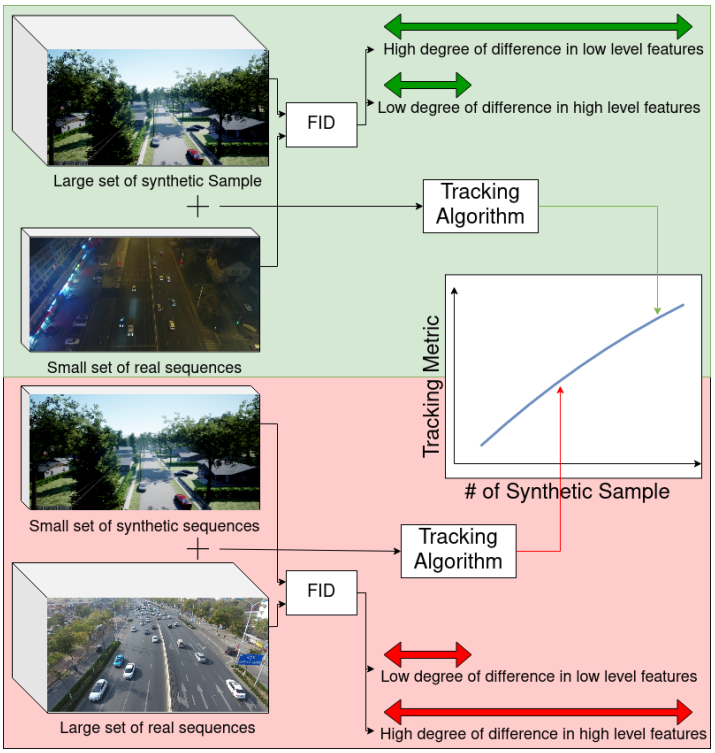


Fig. 1: A combination of synthetic and real datasets with more synthetic samples and higher variance in distribution (top) outperforming another combination with a lower number of synthetic samples and lower variance in distribution (bottom).

level features which makes us inquisitive about the impact of these clusters on the performance of the models, if affected then by how much and why? Next, we examine the impact of different concentrations of photo-realistic sequences on training for the three benchmarks and two rendered datasets one of which is generated by us using a game engine. We demonstrate that the use of synthetic images during training can positively affect performance. Also, we discuss instances where the clusters from our preliminary analysis provide an additional stimulus in the form of a gain or drop in performance. We quantify the variation and provide design guidelines for creating synthetic datasets used to train object-tracking models.

studies. For instance, the Virtual KITTI dataset [9] involves a two-step process where pre-training is performed on the virtual data followed by fine-tuning on real data. The MOTSynth challenge [8] encourages training on synthetic data only and testing on real data, without any use of the latter during the training phase. In contrast, our approach involves fine-tuning models on an amalgamation of both actual and synthetic data, thereby improving tracking performance. To the best of our knowledge, this technique has not been previously explored in the literature.

In this paper, we present the following contributions:

– We investigate the efficacy of integrating synthetic data with real data for improving the performance of Multiple Object Tracking.
– We conduct a comprehensive analysis of our experimental results and offer insights on the synergistic effects of using synthetic and real data. Drawing from our observations, we formulate a set of general recommendations for the generation and incorporation of synthetic data to enhance model performance.

In the following sections, we conduct a literature review of previous works involving synthetic data and Multiple Object Tracking, followed by an analysis of the similarity between real and synthetic data, our experimentation on different datasets and finally a discussion of the results and conclusions drawn from it.

2 Related Works

This section delves into the various methodologies employed by researchers to generate synthetic data for computer vision tasks. The discussion highlights the value of synthetic data in improving deep learning models' performance through techniques such as domain adaptation and pre-training. Additionally, we examine the different tracking techniques that incorporate various concepts, including behavioural models, graph models, convolutional architectures, and transformers, to achieve robust and accurate tracking performance.

2.1 Synthetic Data

To study the impact of synthetic data, the creation of the dataset is important. [13], [24], [25], [33] use RAGE for generating their corresponding virtual datasets. Detouring [6] was employed in [24] to create synthetic a benchmark from commercial software and evaluate visual perception tasks. In [3] a dataset of virtual human subjects under different illumination conditions was developed using Unreal Engine. In [27], Unity Engine was used to develop a dataset for semantic segmentation. Some of the known publically available virtual datasets include Synthia[27] a collection of synthetic images in an urban environment of a virtual city, MOTSynth [8] a large open-source synthetic dataset for pedestrian detection and tracking, Virtual KITTI dataset[9] a synthetic adaptation of popular KITTI Vision benchmark [10].

Various studies demonstrate the efficacy of using synthetic data for enhancing the performance of deep learning models in various computer vision tasks. In [13] utilized only synthetic images to train their model, which outperformed the model trained on actual images in object classification. Wang et al.[33] simulated a crowd in their GCC dataset and proposed the SSIM Embedding cycle GAN for counting crowds in the wild. Sindagi et al.[30] demonstrated that their Gaussian Process-based framework, which was trained on synthetic data, outperformed other domain adaptation techniques that relied only on real data. H. Zunair and A. Hamza [38], utilized domain adaptation to generate synthetic chest X-ray scans and showed that when used as supplementary data during training, the performance of convolutional architectures for classification improved. In [3] a synthetic dataset was used along domain adaptation to improve the performance of a person re-identification task.

2.2 Multiple Object Tracking

L. Taixé et al. introduced a tracker that uses social and grouping behaviours inside a graph model formulating the tracking as a minimum cost flow optimization problem[14]. H. Nam and B. Han proposed MDNet[20] a multi-domain learning convolutional neural network framework that learns domain-independent features during pretraining and domain-specific information during the tracking.

L. Bertinetto[5] trained a fully convolutional Siamese network to learn a similarity function in an offline manner to be evaluated online during training to locate a template image within the search image using the strong embeddings learned in the offline phase. B. Li[15] proposed a Siamese Region Proposal Network consisting of a template and a detection branch which are trained offline and correlational maps for feature extraction, the tracking is formulated as a local one-shot detection task.

P. Bergman and T. Meinhardt introduced tracktor[4] that exploits bounding box regression of an existing object detector, without any additional training required for tracking objects. Zhou presented a point-based tracking framework called CenterTrack[35] that uses CenterNet[36] detector conditioned on two consecutive frames that also predicts a displacement vector for associating positions of the objects through frames.

Y.Zhang et al. introduced FairMOT[34], also a CenterNet[36] based technique for a multi-task learning approach for detection and re-identification. In this technique, the competition for accuracy between the two tasks was addressed by introducing fairness. This results in an unbiassed network which treats both tasks equally thus, it doesn't affect their accuracy adversely.

G. Ning et. al proposed ROLO[21] a recurrent extension of YOLO[22] architecture by adding an LSTM stage, training involves three phases pertaining of convolutional layers, training of object proposal module and training of recurrent LSTM module.

P. Sun et al. introduced TransTrack[31] an attention-based query-key scheme inspired by transformers[32] that uses attention to track objects, their framework generates two sets of queries containing information for new coming objects and information for maintaining tracklets. T. Meinhardt et al. introduced Trackformer[18] following the tracking by attention paradigm for joint detection and tracking, attention is computed between frame features, tracks and object queries to output bounding boxes and identities.

3 Synthetic Dataset & Observations

In this section we touch upon the synthetic dataset that we created and a publically available synthetic dataset for supplementing the real dataset. We also discuss our key observations when comparing these synthetic datasets to real video sequences using Fréchet Inception Distance (FID) [12].

Our experimental setup utilizes synthetic video sequences generated with the AirSim plugin [29] for Unreal engine. Details on the generation of the dataset are discussed in Section 4.1. Along with this new dataset, we use two published Unmanned Aerial Vehicle (UAV) benchmarks for the detection and tracking of vehicles, UAVDT benchmark [7] and VisDrone dataset [37] as real sequences. To eliminate bias due to the domain and task, and ensure the generalization of the insights, we additionally use another pair of real and synthetic tracking datasets with people tracking in place of vehicle tracking as the objective. For this purpose, we use MOT17 [19] and MOTSynth [8] as real and synthetic datasets respectively.

Fréchet Inception Distance (FID) is a quality measure first introduced in [12], for capturing the similarity of the images generated by GANs, this metric also correlates with human judgement. FID score for 2 identical images is 0, and for 2 identical sets of images or videos is close to 0. It increases as the visual similarities between the two images or sets of images reduce as depicted in Figure 2. A synthetic sequence with similar lighting, camera angle and elevation as the real sequence results in a relatively lower FID in contrast to another real sequence that has different lighting, camera angle and elevation. We use it to estimate the degree of similarity between synthetic and real images for each pair of real and synthetic sequences.

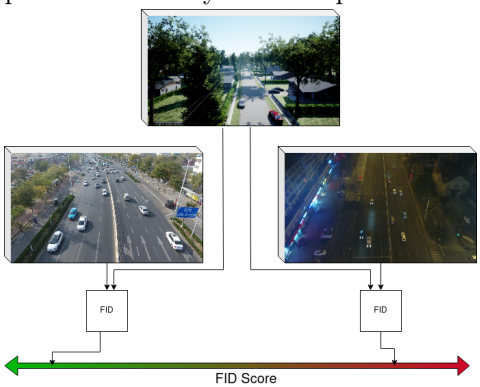


Fig. 2: FID computation example for low level features between AirSim generated video sequence (top) with a similar looking (bottom-left) and a different looking (bottom-right) real video sequence from UAVDT dataset. The computed value towards the green end depicts similarity between the two sequences, and contrary to that towards the red side depicts visual dissimilarity.

We computed the Fréchet Inception Distance for three combinations of real and synthetic datasets from the first pooling layer features (FID64), the second max pooling features (FID192), the pre-auxiliary classifier features (FID768) and the final average pooling features (FID2048)[28]. The computations from the second max pooling (FID192) and the final average pooling (FID2048) features do not add more information or echoes that the first max pooling features (FID64) and the pre-auxiliary

classifier features (FID768) already express. Thus, we only use FID scores obtained from the first max pool layer features and the pre-auxiliary classifier features for the low level features and the high level features respectively. The FIDs for all synthetic and real datasets are plotted as heatmaps in Figure 3.

In the heatmaps, each row is a real sequence and each column is a synthetic sequence from corresponding real and synthetic dataset pairs. Within the heat maps for low level features 3a, the combination of UAVDT benchmark and AirSim generated dataset shows patterns of higher FID scores for some real sequences while most of the real sequences have a relatively lower FID Score. The other 2 dataset combinations only observe a relatively low FID score for all real and synthetic sequence pairings. Contrasting to this, the heatmaps for high level features 3b, show more patterns of high FID scores for real sequences in all three dataset combinations. Interestingly, a pattern for high FID score is also noticeable for synthetic sequences in the MOT17 and MotSynth datasets pairing. We discuss these patterns further in this section.

These initial observations from the heatmaps motivate us to define rigid clusters based on the FID computations. We cluster sequences under 3 categories namely, lower, moderate or higher degrees of difference in low or high level features. This clustering is required to isolate features on the basis of similarity and measure their impact on the performance of the training process. We later use these clusters in further sections for experimentation and discussion.

For the clustering, we create a range between 0 (the minimum achievable distance) and the maximum calculated FID determined across all datasets plus an additional buffer. In our experiments the max value for low level and high level featres were 45 and 3 with additional buffers of 5 and 0.5. We fit all the sequences to this range and scale them to get a range between 0 and 1. This scale is divided into three parts using 0.3 and 0.6 as the division points. The sequences that fall under the first, the second and the third segment are termed as sequences with lower, moderate or higher degrees of difference respectively. Although this work uses FID as a measure for calculating similarity, the use of other metrics is also encouraged. FID was used because of the demerits of other metrics highlighted in [12].

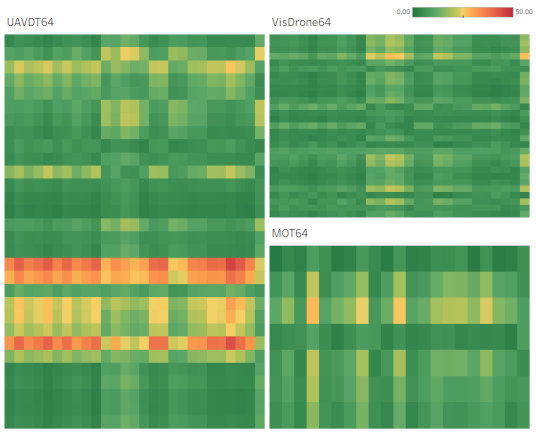
3.1 FID for low level features

The FIDs obtained after first pooling layer features for the sequences from the UAVDT benchmark and the AirSim generated dataset, sequences with a higher degree of difference have an average of 32.80 $\pm$2.60, the same for sequences having a moderate degree of difference is 18.93 $\pm$2.33 and finally the sequences with a lower degree of difference have an average of 7.80 $\pm$2.93. We observe that all sequences in the VisDrone dataset compared to the synthetic dataset generated using AirSim have a lower degree of difference for low level features with an average of 7.40 $\pm$2.31. A similar observation is made for the sequences from the MOT17 and the MOTSynth datasets, all the sequences lead to an average of 8.66 $\pm$3.48 thus falling under a lower degree of difference for low level features. The FIDs for all synthetic and real sequences are visually represented in Figure 3a, where the green regions represent the lower degree of difference, yellow-orange shades depict the moderate degree of difference and dark orange-red represents the higher degree of difference.

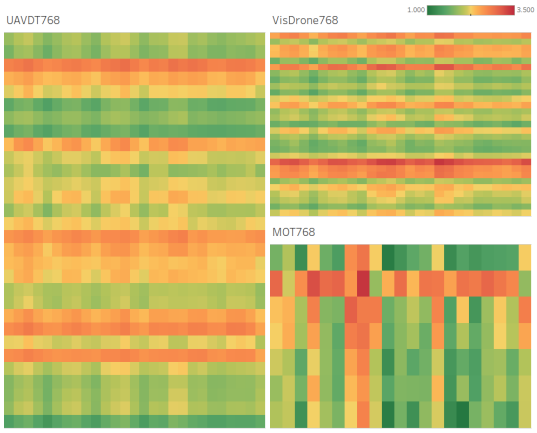
3.2 FID for high level features

On the basis of the clustering scheme discussed earlier in this section and the FIDs obtained from the pre-auxiliary classifier features for UAVDT and AirSim generated sequences, we cluster sequences among low, medium or high degrees of difference for high level features among real data. The average values for clusters are 1.60 $\pm$0.05, 2.13 $\pm$0.19 and 2.65 $\pm$0.12 respectively. With the same synthetic dataset when FID values are calculated along the VisDrone dataset the clusters obtained have 1.72 $\pm$0.01 for lower degree of difference, 2.07 $\pm$0.22 for a moderate degree of difference and 2.76 $\pm$0.21 for a higher degree of difference in high level features. When calculating the FIDs with higher level features

for MOT17 and MOTSynth datasets, we obtain 1.75, 1.99 ±0.16 and 2.72 for lower, moderate and higher degrees of differences. Unlike other real and synthetic dataset combinations, we also observe a pattern for synthetic sequences for MOT17 and MOTSynth. The average values for lower, moderate and higher degrees of difference in high level features are 1.57 ±0.09, 2.01 ±0.20 and 2.68 ±0.18 respectively. We visualize these FID computations in the form of a heatmap in Figure 3b.



(a) FID64: Fréchet Inception Distance obtained from first pooling features. Left: UAVDT and AirSim, Right-Top: Visdrone and AirSim, Right-Bottom: MOT and MOTSynth. Each row represents a real sequence and each column represents a synthetic sequence.



(b) FID768: FID obtained from pre-auxiliary classifier features. Left: UAVDT and AirSim, Right-Top: Visdrone and AirSim, Right-Bottom: MOT and MOTSynth. Each row represents a real sequence and each column represents a synthetic sequence.

Fig. 3: FIDs heatmaps for low level features (a) and high level features (b). Dark green represents lower degree of difference, dark red represents higher degree of difference, and the shades in between represent moderate degree of difference in lower/higher level features.

3.3 Objectives

With the derived insights and our objective of impact investigation of synthetic data as a supplement in combination with real data, we aim to answer the following questions:

- How effective is the use of synthetic data when supplementing a real dataset?
- What is the impact of different real-synthetic concentrations on the performance metric?
- What are the characteristics of the synthetic data that drive this change, the concentration, the degree of diversity in information brought by the synthetic samples or both factors?

4 Experiments

In this section, we discuss in detail the synthetic and the real datasets that we use, our strategy to answer the questions that were raised in the previous section and our trials.

4.1 Datasets

As already discussed briefly in Section 3, we generate a set of synthetic video sequences using the AirSim plugin in Unreal engine. To generate the simulated video sequences, we load the environment with a simulated drone and dictate its flight trajectory by a set of three-dimensional points in the simulation environment transmitted through APIs. We assimilate input from the camera mounted on

the virtual drone and detect vehicles within the field of view of the drone using another pair of APIs to save frames and annotation to a local storage device. We also alter weather conditions across different flight paths to create a diverse set of simulated video sequences. In total, we generated 25 sequences exhibiting different weather conditions, providing a diverse range of scenarios for evaluation.

For the real dataset we use the Unmanned Aerial Vehicle Benchmark [7] (UAVDT) which is a collection of video sequences captured by drones. This benchmark dataset offers sequences with various conditions for illumination, camera viewpoint and elevation. To ensure that our experimentation is not limited to a single dataset, we also conduct tests on another UAV detection and tracking dataset called VisDrone [37]. It contains both city and country environments with annotations for many objects in various weather and lighting conditions. Since our synthetic dataset only had information about vehicles, we rank all the sequences on the most number of vehicles in the scene and only considered the top 30 videos which had the most number of vehicles for our vehicle tracking experiments.

For the extensiveness of our experiments, we use another pair of real and synthetic datasets. MOT17 [19] dataset which is a pedestrian detection and tracking dataset with video sequences having different viewports, camera movements and weather conditions. For the synthetic part, we used the MOTSynth [8] dataset which was created for pedestrian detection, tracking and segmentation and contains frames generated using a rendering game engine. We only required a limited number of sequences according to our experiment setup and a random selection of 21 sequences is used to serve as the training set.

4.2 Strategy

We use models trained only on real datasets as baseline models to compare and evaluate against the results obtained from models discussed further in our training strategy. In our strategy, we keep the total number of real and synthetic training sequences constant, that is the number of real sequences available for training. We then substitute real sequences with synthetic sequences. We focus on the substitution and not on the addition of new data for two reasons. First, additional training data will lead to unfair evaluation as the new dataset will have more training samples when compared to the baseline model. Second, substitution creates an artificial scarcity of data enabling us to evaluate the impact of synthetic data when the actual data is insufficient or missing. For these reasons, we formulate an approach to break down the datasets into different-sized folds such that, a bigger chunk from the real dataset has a complementary smaller fold in the synthetic dataset and vice-versa. The combined dataset always accounts for the same number of total video sequences as originally in the training set for the real dataset. For the vehicle tracking experiments, we use ratios 1:5, 1:2, 1:1, 2:1 and 5:1 between real and synthetic data i.e. when there are 5 real sequences we use 25 synthetic sequences, 10 real and 20 synthetic sequences and so on. For people tracking experiments, we use 1:6, 1:2, 2:1 and 6:1 as the ratios.

We use multiple folds for each concentration of real-synthetic combination to understand the consistency of the change in the tracking metric with the real-to-synthetic data ratio. Within the folds, we vary the number of sequences with lower, moderate and higher degrees of difference for low-level and high-level features as discussed in the previous section. In our experiments, each fold is denoted by a lowercase letter.

4.3 Training

For our experiments, we train FRCNN[23] network for object detection and ResNet50 [11] model for re-identification, together these two models are used in combination as described in the Tracktor[4] technique. The datasets, both real and synthetic are aimed for detection and tracking and not for re-identification. To allow training of re-identification models on these sequences we create a re-identification dataset from the given frames. We crop the frames where bounding boxes are present and

use these crops for tracked objects as a re-identification dataset. We use the described setup of an object detector and a re-identifier with different folds of the training set as discussed in Section 4.2. Models trained on different folds are evaluated using HOTA[17] and IDF1 [26] as the calculative measures for assessing performance. The metrics IDP and IDR are intermediatory measures that are needed to calculate IDF1 value while DetA and AssA are used to calculate HOTA. The metrics are calculated using trackeval [16]. The results of public detection from different manifestations of the Tracktor on the UAVDT benchmark and the AirSim generated dataset are reported in Table 4, on VisDrone dataset and AirSim generated dataset are reported in Table 5 and on MOT17 and MOTSynth are reported in Table 6. Further, we discuss these results in Section 5.

Also, we extend the applicability of synthetic datasets to transformer-based architectures. We select the Transtrack [31] architecture, which is an encoder-decoder framework with a ResNet-50 [11] backbone network. We train the models on MOT17 and MOTSynth datasets, using the same concentrations and folds as used for the Tracktor experiments. Results are reported in Table 7 and further discussed in Section 5.

5 Discussion

Tables 4 and 6 show a significant increase in performance measure when synthetic data is included in the training set against the benchmark that only contains all real data. The improvement is up to a 14% increase for the UAVDT benchmark and up to a 10% increase for the MOT17 dataset. Also, Table 5 shows an increase up to 4% was achieved for the VisDrone dataset. Table 7 shows an increase up to 7% for the MOT17 dataset on a transformer-based architecture. There is a positive correlation between the percentage of synthetic data in the training set and the performance measure for the UAVDT benchmark and the VisDrone dataset. The performance increase for the MOT17 dataset is moreover constant and is not affected by changes in dataset concentrations on Tracktor but we again observe the correlation between the number of synthetic samples and the tracking metric on transformer-based architecture. We further discuss each dataset individually under the following subsections.

Real Set		Synthetic Set							
Size	Fold	Size	Fold	IDP↑	IDR↑	DetA↑	AssA↑	IDF1↑	HOTA↑
5	a	25	a	84.232	83.500	**77.950**	63.984	83.864	70.489
5	b	25	a	84.877	83.538	75.616	63.979	84.202	69.405
5	c	25	a	78.229	78.535	67.812	59.057	78.382	63.134
5	d	25	a	**90.251**	**87.227**	77.166	**70.266**	**88.713**	**73.504**
5	e	25	a	85.501	84.279	72.327	62.985	84.885	67.313
5	f	25	a	85.632	83.054	76.823	64.676	84.323	70.378
10	g	20	b	88.740	82.318	73.588	67.232	85.409	70.252
10	h	20	b	82.459	81.612	68.126	62.729	82.033	65.235
10	i	20	b	87.303	81.945	69.367	64.190	84.539	66.584
15	j	15	c	90.029	79.524	66.662	66.109	84.451	66.259
15	k	15	c	87.484	80.520	67.165	63.453	83.858	65.163
20	l	10	d	81.393	78.686	64.573	61.158	80.016	62.722
20	m	10	e	89.997	76.785	62.950	65.291	82.868	64.004
20	n	10	f	85.122	75.669	62.724	63.897	80.118	63.146
25	o	5	g	81.287	75.073	60.347	60.683	78.057	60.378
25	p	5	h	82.752	77.801	62.214	61.078	80.200	61.513
25	q	5	i	81.746	78.741	64.828	61.166	80.216	62.832
25	r	5	g	86.904	69.093	54.706	62.014	76.982	58.104
25	s	5	h	83.274	74.501	61.816	62.026	78.644	61.744
25	t	5	i	82.871	78.444	63.305	61.988	80.597	62.432
30	u	0	NA	82.085	75.486	61.183	60.948	78.647	60.921

Fig. 4: Results for Tracktor technique trained on datasets with different concentrations of UAVDT benchmark (real) and AirSim generated dataset (synthetic). Column Size denotes the number of sequences and Fold denotes which fold was used.

Real Set		Synthetic Set							
Size	Fold	Size	Fold	IDP↑	IDR↑	DetA↑	AssA↑	IDF1↑	HOTA↑
5	a	25	a	69.957	72.991	64.611	**57.841**	71.442	**60.701**
5	b	25	a	63.117	65.636	63.323	50.193	64.352	55.975
5	c	25	a	68.029	70.809	59.921	54.752	69.391	56.779
5	d	25	a	68.278	70.847	**65.364**	55.616	69.539	59.961
5	e	25	a	65.193	67.949	64.281	52.798	66.543	57.834
5	f	25	a	71.052	**74.107**	61.694	57.558	**72.547**	59.241
10	g	20	b	63.448	66.078	63.482	50.621	64.736	56.271
10	h	20	b	69.845	72.378	62.859	56.169	71.089	59.043
10	i	20	b	67.547	70.272	63.569	54.147	68.882	58.307
15	j	15	c	66.078	68.449	61.602	51.998	67.242	56.239
15	k	15	c	67.924	70.180	64.041	54.747	69.034	58.902
20	l	10	d	65.904	67.146	61.679	51.545	66.519	56.053
20	m	10	e	65.603	67.782	64.136	51.997	66.675	57.462
20	n	10	f	67.121	68.683	63.215	53.425	67.893	57.779
25	o	5	g	65.085	65.854	62.107	50.922	65.467	55.953
25	p	5	h	65.430	66.037	61.531	50.750	65.732	55.557
25	q	5	i	65.541	66.822	62.856	51.431	66.176	56.511
25	r	5	g	69.491	67.100	62.294	52.595	68.275	56.943
25	s	5	h	**72.598**	71.483	62.676	55.977	72.036	58.948
25	t	5	i	64.689	65.665	62.643	50.480	65.173	55.885
30	u	0	NA	64.324	64.601	61.910	49.864	64.462	55.252

Fig. 5: Results for Tracktor technique trained on datasets with different concentrations of Visdrone dataset (real) and AirSim generated dataset (synthetic). Column Size denotes the number of sequences and Fold denotes which fold was used.

5.1 UAVDT

We observe a direct link between the performance measure and the percentage of the synthetic dataset in the overall training set, by increasing the number of synthetic samples we notice an increase in the HOTA metric. It is highest when we use twenty-five synthetic samples and five real ones, and lowest when use twenty-five real and five synthetic samples across a number of folds. Also, among folds comprised of five real and twenty-five synthetic sequences, the HOTA metric is highest when the training set includes sequences with a higher degree of difference for low level features. Additionally, in the folds consisting of twenty-five real and five synthetic sequences, we notice that the HOTA metric reduces when the folds are constituted from sequences with low or moderate degrees of difference for low level features. All experiments are reported in Table 4.

5.2 VisDrone

Our findings indicate an increasing trend, albeit with a few deviations for the VisDrone and AirSim datasets. The performance of models trained on different folds generally increases, except for the folds where five sequences are synthetic the models perform worst than the benchmark but the performance increases gradually as the percentage of synthetic data increases. Another deviation is remarked, where models trained on folds with 20 synthetic sequences perform better than the models trained on folds with synthetic data but the latter still outperforms the benchmark model.

With the FIDs analysis for low level features (Section 3.1) as the foundation, it is hard to come to conclusions as all sequences for this dataset combination fall under a low degree of difference. Drawing on the insights derived from the FID analysis for high level features (Section 3.2), experiments with folds having 5, 10 or 15 real sequences, the fold having the most number of sequences with a lower degree of difference for high level features outperforms the rest of the folds in that category.

Real Set		Synthetic Set							
Size	Fold	Size	Fold	IDP↑	IDR↑	DetA↑	AssA↑	IDF1↑	HOTA↑
3	a	18	a	48.893	63.291	40.617	52.277	55.168	45.904
3	b	18	b	49.996	59.487	41.928	48.766	54.330	45.069
3	c	18	c	53.866	59.708	**44.196**	49.285	56.636	46.547
3	d	18	d	**55.581**	62.286	43.842	51.946	**58.743**	47.559
3	e	18	e	48.792	61.824	41.173	50.731	54.540	45.552
3	f	18	f	50.057	64.044	40.420	53.327	56.193	46.312
3	g	18	g	52.789	61.614	43.340	50.591	56.861	46.726
7	h	14	h	46.250	61.112	39.961	49.537	52.652	44.250
7	i	14	i	52.175	63.552	42.574	52.785	57.304	47.294
7	j	14	j	48.708	62.582	41.035	52.501	54.780	46.313
14	k	7	k	47.634	62.714	40.470	50.411	54.143	45.080
14	l	7	l	49.877	65.935	41.136	**56.227**	56.793	**47.927**
14	m	7	m	50.716	63.499	42.412	53.109	56.392	47.333
18	n	3	n	48.730	63.345	40.822	52.392	55.084	46.142
18	o	3	o	50.245	64.282	41.520	53.084	56.403	46.873
18	p	3	p	48.210	63.303	40.821	52.738	54.735	46.289
18	q	3	q	50.598	**66.076**	41.154	56.023	57.311	47.906
18	r	3	r	49.462	65.915	41.347	55.496	56.515	47.806
18	s	3	s	50.404	65.135	41.934	54.193	56.830	47.570
18	t	3	t	50.398	63.233	41.706	53.768	56.090	47.234
21	u	0	NA	34.597	46.205	40.230	32.198	39.567	35.883

Fig. 6: Results for Tracktor technique trained on datasets with different concentrations of MOT17 dataset (real) and MOTSynth dataset (synthetic). Column Size denotes the number of sequences and Fold denotes which fold was used.

Real Set		Synthetic Set							
Size	Fold	Size	Fold	IDP↑	IDR↑	DetA↑	AssA↑	IDF1↑	HOTA↑
3	a	18	a	62.653	53.443	45.824	49.286	57.683	46.968
3	b	18	b	44.166	72.288	39.513	67.894	54.831	51.514
3	c	18	c	66.111	57.044	47.974	51.432	61.244	49.268
3	d	18	d	67.937	55.626	45.120	53.142	61.168	48.695
3	e	18	e	67.314	64.160	49.847	55.861	65.699	52.523
3	f	18	f	**74.162**	59.827	**51.015**	56.889	**66.228**	53.463
3	g	18	g	62.410	57.089	43.517	53.589	59.631	47.928
7	h	14	h	44.227	73.213	40.035	67.432	55.143	51.738
7	i	14	i	62.208	64.166	48.555	54.669	63.172	51.261
7	j	14	j	63.459	66.120	49.082	58.896	64.762	53.421
14	k	7	k	43.836	78.055	40.616	68.205	56.142	52.385
14	l	7	l	45.196	79.390	41.596	68.821	57.600	53.242
14	m	7	m	60.775	67.998	48.565	59.332	64.184	53.390
18	n	3	n	44.370	78.334	41.759	68.752	56.651	53.378
18	o	3	o	42.178	77.860	40.276	66.858	54.716	51.656
18	p	3	p	44.261	77.375	40.184	69.770	56.311	52.751
18	q	3	q	43.894	78.164	41.590	66.904	56.218	52.510
18	r	3	r	43.388	79.442	40.974	69.318	56.124	53.073
18	s	3	s	60.466	59.020	44.717	56.860	59.734	50.124
18	t	3	t	44.689	**81.538**	41.186	**69.973**	57.735	**53.523**
21	u	0	NA	43.207	79.424	41.177	67.884	55.967	52.633

Fig. 7: Results for TransTrack architecture trained on datasets with different concentrations of MOT17 dataset (real) and MOTSynth dataset (synthetic). Column Size denotes the number of sequences and Fold denotes which fold was used.

5.3 MOT17

We observe up to 10% increase in HOTA metric for supplementing the MOT17 dataset with the MOTSynth dataset. However, unlike the previous two benchmarks where an increasing trend was

observed, our examination reveal about a constant increase in performance measure invariant of the real-synthetic concentrations throughout the experiments.

Guided by the FIDs analysis of high level features, amongst the folds with 3 real sequences, the HOTA metric is the least when trained on samples with higher degree of difference for high level features in comparison to when these samples are excluded. The same is observed in folds with 7 real sequences. This phenomenon becomes hazy for folds with 18 real scenarios. Also to note, the performance metric improves when sequences with higher degree of difference are excluded from the folds comprising synthetic data.

Experiments with the TransTrack architecture show a similar result, an almost constant trend for the HOTA metric. However, the trend is clearly visible in the IDF1 metric. The performance of the model is directly correlated with the amount of synthetic data in the training dataset. The fold including the sequences with a higher degree of difference for high-level features, always performs the worst among the folds of the same size.

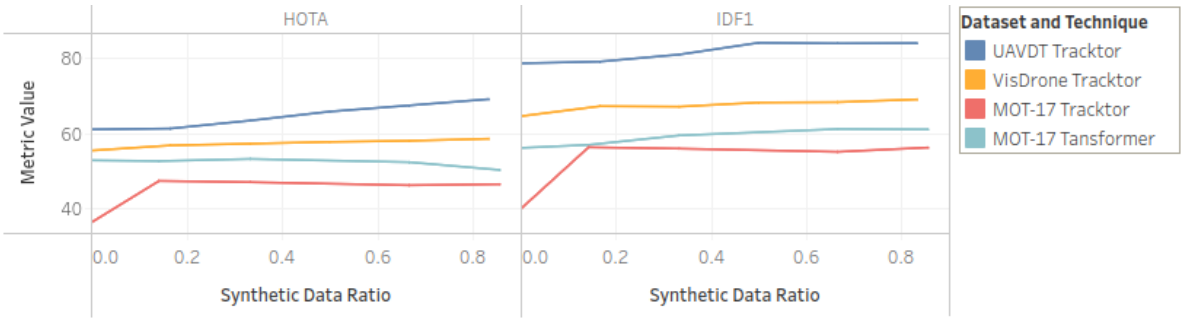


Fig. 8: Change in HOTA and IDF1 measures on increasing the concentration of synthetic samples in the training set for Tracktor on UAVDT benchmark, VisDrone dataset and MOT17 dataset; for TransTrack on MOT17 dataset.

5.4 Guidelines

Considering the key insights derived in this section we can deduce that by using synthetic data, one can increase the performance of a model. We derive the following principles from our experiments.

- When synthetic data is used in orders of magnitudes of real data a performance can be anticipated. In our experiments, the increase in performance was up to 15% when synthetic data accounted five times more than the actual video sequences.
- The performance improvement is higher when the variance in low-level features is high. In our experiments, we clustered sequences with values greater than 0.6 on our scale (FIDs greater than 30 units calculated from first pooling layer features) as a high degree of difference for low-level features. The presence of these sequences resulted in a better performance.
- The increase in performance is limited by the variance in high-level features and is recommended to be kept minimal. Our experiments with sequences with values lower than 0.3 on our scale(under 1.75 units for FIDs calculated from pre-auxiliary classifier features) showed an increased improvement.

6 Conclusion

In this study, we investigated the effectiveness of using synthetic data in combination with real data for Single Camera Multi-Object Tracking tasks. We utilized three different datasets and two different

tracking techniques to evaluate the impact of using synthetic data. Our results indicate that the inclusion of synthetic data in the training process of deep learning models improves the performance metrics when compared to using real data alone. Furthermore, we also explored the specific aspects of synthetic data that should be emphasized to further enhance the performance of the models.

Our findings suggest that the combination of real and synthetic data can lead to a new paradigm for training deep learning models. While synthetic data has traditionally been used for pre-training or domain adaptation, our study highlights the potential for a simpler technique to complement real data in the training process. We aim to validate the application of synthetic data for solving challenges such as bias mitigation, generalization of outside datasets, and wider applicability of existing datasets in our future works. We believe that this approach can lead to improved performance in a range of computer vision tasks and can pave the way for the development of more sophisticated and accurate models. Overall, this paper contributes to the growing body of research on the use of synthetic data and its potential for enhancing the capabilities of deep learning models.

Acknowledgement

This work is financially supported by the Natural Sciences and Engineering Research Council of Canada Grants RGPIN-2021-03479 (NSERC DG) and ALLRP 571887 - 2021 (NSERC Alliance).The authors would like to thank Sacha Leprêtre (CAE, formerly Presagis Inc, Canada) and his team for their continued support of this work.

References

1. Adimoolam, Y.K., Chatterjee, B., Poullis, C., Averkiou, M.: Efficient deduplication and leakage detection in large scale image datasets with a focus on the crowdai mapping challenge dataset. arXiv preprint arXiv:2304.02296 (2023)
2. Baek, K., Shim, H.: Commonality in natural images rescues gans: Pretraining gans with generic and privacy-free synthetic data. In: Proceedings of the IEEE/CVF CVPR. pp. 7854–7864 (2022)
3. Bak, S., Carr, P., Lalonde, J.: Domain adaptation through synthesis for unsupervised person re-identification. CoRR **abs/1804.10094** (2018), http://arxiv.org/abs/1804.10094
4. Bergmann, P., Meinhardt, T., Leal-Taixé, L.: Tracking without bells and whistles. In: ICCV (2019)
5. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional siamese networks for object tracking. CoRR **abs/1606.09549** (2016), http://arxiv.org/abs/1606.09549
6. Brubacher, D.: Detours: Binary interception of win32 functions. In: Windows NT 3rd Symposium (Windows NT 3rd Symposium). USENIX Association, Seattle, WA (Jul 1999)
7. Du, D., Qi, Y., Yu, H., Yang, Y., Duan, K., Li, G., Zhang, W., Huang, Q., Tian, Q.: The unmanned aerial vehicle benchmark: Object detection and tracking. In: ECCV (2018)
8. Fabbri, M., Brasó, G., Maugeri, G., Cetintas, O., Gasparini, R., Ošep, A., Calderara, S., Leal-Taixé, L., Cucchiara, R.: Motsynth: How can synthetic data help pedestrian detection and tracking? In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10849–10859 (2021)
9. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtual worlds as proxy for multi-object tracking analysis. CoRR **abs/1605.06457** (2016), http://arxiv.org/abs/1605.06457
10. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR (2012)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015), http://arxiv.org/abs/1512.03385
12. 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. NeurIPS (2017)

13. Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S.N., Vasudevan, R.: Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? CoRR **abs/1610.01983** (2016), http://arxiv.org/abs/1610.01983
14. Leal-Taixé, L., Pons-Moll, G., Rosenhahn, B.: Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In: ICCV Workshops. pp. 120–127 (2011). https://doi.org/10.1109/ICCVW.2011.6130233
15. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: 2018 IEEE/CVF CVPR. pp. 8971–8980 (2018). https://doi.org/10.1109/CVPR.2018.00935
16. Luiten, J., Hoffhues, A.: Trackeval. https://github.com/JonathonLuiten/TrackEval (2020)
17. Luiten, J., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., Leibe, B.: Hota: A higher order metric for evaluating multi-object tracking. International Journal of Computer Vision pp. 1–31 (2020)
18. Meinhardt, T., Kirillov, A., Leal-Taixe, L., Feichtenhofer, C.: Trackformer: Multi-object tracking with transformers. In: IEEE CVPR (June 2022)
19. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: MOT16: A benchmark for multi-object tracking. arXiv:1603.00831 [cs] (Mar 2016), http://arxiv.org/abs/1603.00831, arXiv: 1603.00831
20. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. CoRR **abs/1510.07945** (2015), http://arxiv.org/abs/1510.07945
21. Ning, G., Zhang, Z., Huang, C., Ren, X., Wang, H., Cai, C., He, Z.: Spatially supervised recurrent convolutional neural networks for visual object tracking. In: 2017 IEEE ISCAS. pp. 1–4. IEEE (2017)
22. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
23. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. CoRR **abs/1506.01497** (2015), http://arxiv.org/abs/1506.01497
24. Richter, S.R., Hayder, Z., Koltun, V.: Playing for benchmarks. CoRR **abs/1709.07322** (2017), http://arxiv.org/abs/1709.07322
25. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. CoRR **abs/1608.02192** (2016), http://arxiv.org/abs/1608.02192
26. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: Computer Vision–ECCV. pp. 17–35. Springer (2016)
27. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: CVPR. pp. 3234–3243 (2016). https://doi.org/10.1109/CVPR.2016.352
28. Seitzer, M.: pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid (August 2020)
29. Shah, S., Dey, D., Lovett, C., Kapoor, A.: Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and Service Robotics (2017), https://arxiv.org/abs/1705.05065
30. Sindagi, V.A., Yasarla, R., Babu, D.S., Babu, R.V., Patel, V.M.: Learning to count in the crowd from limited labeled data. In: European Conference on Computer Vision. pp. 212–229. Springer (2020)
31. Sun, P., Cao, J., Jiang, Y., Zhang, R., Xie, E., Yuan, Z., Wang, C., Luo, P.: Transtrack: Multiple-object tracking with transformer. arXiv preprint arXiv: 2012.15460 (2020)
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS **30** (2017)
33. Wang, Q., Gao, J., Lin, W., Yuan, Y.: Learning from synthetic data for crowd counting in the wild. In: IEEE CVPR. pp. 8198–8207 (2019)
34. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: Fairmot: On the fairness of detection and re-identification in multiple object tracking. International Journal of Computer Vision **129**, 3069–3087 (2021)
35. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. ECCV (2020)
36. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)
37. Zhu, P., Wen, L., Du, D., Bian, X., Fan, H., Hu, Q., Ling, H.: Detection and tracking meet drones challenge. IEEE TPAMI pp. 1–1 (2021). https://doi.org/10.1109/TPAMI.2021.3119563
38. Zunair, H., Hamza, A.B.: Synthesis of covid-19 chest x-rays using unpaired image-to-image translation. Social network analysis and mining **11**, 1–12 (2021)