Scaling up Drone Detection using Synthetic Data

David Novikov[1] and Rohit Gupta[2] and Mubarak Shah[2]

Abstract— **As drones become increasingly ubiquitous, they will require the ability to detect airborne objects in order to achieve full autonomy in the wild. In recent years, labeled datasets for drone detection have emerged, and have been used to train object detection models to solve this problem. However, collecting such datasets is unduly expensive, and typically each dataset is limited to a single location. The advent of larger and deeper neural networks also necessitates scaling up training datasets. We propose to train drone detectors utilizing synthetic data which reflects a diverse variety of situations where drones might potentially be utilized. We develop and test two strategies for generating synthetic data: utilizing scraped web images and using a diffusion model pre-trained on web images for guided generation. Both of these methods benefit from a large amount of unlabeled image data available on the web. Using our synthetic data for training, we match or exceed state of the art results on EPFL Drones and NPS Drone detection benchmarks by up to 6% and for the first time demonstrate generalization across datasets from different geographical locations.**

I. INTRODUCTION

Drones are increasingly seeing a variety of real world deployments for a diverse set of applications such as agriculture, delivery, defense, and disaster relief. Existing drones have varying levels of autonomy ranging from fully manual control by remote operator(s) to fully autonomous control. As deployments of drones continue, increasing crowding of the skies, particularly in dense urban contexts, is increasingly becoming an important challenge to deal with in order to achieve full autonomy. Drones need to detect and avoid other drones and other airborne objects to successfully complete their flights. The efficacy of deep learning based detection models for this task has previously been successfully demonstrated in the literature. These methods typically use labeled drone datasets such as EPFL Drones and NPS Drones to train their models. While these models achieve strong performance on the benchmarks, this falls short of demonstrating success that could carry over to the real world since these benchmark datasets are limited in terms of location, conditions, and types of drones used.

In parallel, there have been massive strides in the compute power available to edge devices [1], [2] such as drones. This permits the deployment of bigger and more accurate models for the task of detection. But bigger models require more data for training. However, simply scaling up labeled drone detection datasets is not practical because of the logistics and cost involved. In other domains requiring visual perception, self-supervised and weakly supervised learning using massive amounts of unlabelled online data have been utilized to great success. However, these methods are not directly applicable for recognizing objects like drones which are rare in natural images. The goal of this work is to overcome this limitation and develop a method for training drone detectors which can benefit from the large and growing corpus of online visual data.

We propose a simple fix for this problem: a small dataset of the object of interest (drones) along with a large number of unlabelled background images can be combined using simple copy-paste operation and data augmentations to generate a massive amount of synthetic detection data. This data has the ”ground-truth” detection labels available for free by construction. The availability of this data allows us to train larger detection models without overfitting, while also resulting in better generalization to scenarios not present in existing labelled datasets.

While web image and video data is remarkably diverse, in certain applications, it is desirable to generate synthetic data in a controlled fashion, with guidance provided in some form, such as a description or reference images. We utilize pre-trained latent diffusion models to generate synthetic background images guided by reference samples from an existing dataset. These synthetic background images can then be used for improving the performance of detection models in a transfer learning setting.

Appropriate benchmarks and evaluation protocols that closely match real world conditions are a pre-requisite for measuring progress. We build upon existing benchmarks and propose two new settings to evaluate drone detection models. Existing works generate test-train splits by temporally splitting the flight video sequences in half. We propose a more challenging split where the test and train split consists of independent flight sequences. We also propose evaluating generalization of detection models across datasets collected in different parts of the world.

The contributions of our work can be briefly summarized as follows:

- Novel method for generating and using synthetic data to train drone detection models, exceeding state of the art performance
- Exploration of the effectiveness of latent diffusion models to generate synthetic data guided by existing drone detection datasets
- Two new settings (independent test-train sequences and cross dataset transfer) to evaluate drone detection models to more closely mirror real world conditions

David Novikov was funded by an NSF REU Grant XXX, Rohit Gupta recognizes support from YYY

[1]David Novikov is a student at Ohio State University, this work was done as part of the UCF CRCV NSF-REU Site 2022 dn9678@gmail.com

[2] Rohit Gupta and Mubarak Shah are with the Center for Research in Computer Vision, University of Central Florida, Orlando, FL 32816, USA rohitg@knights.ucf.edu, shah@crcv.ucf.edu

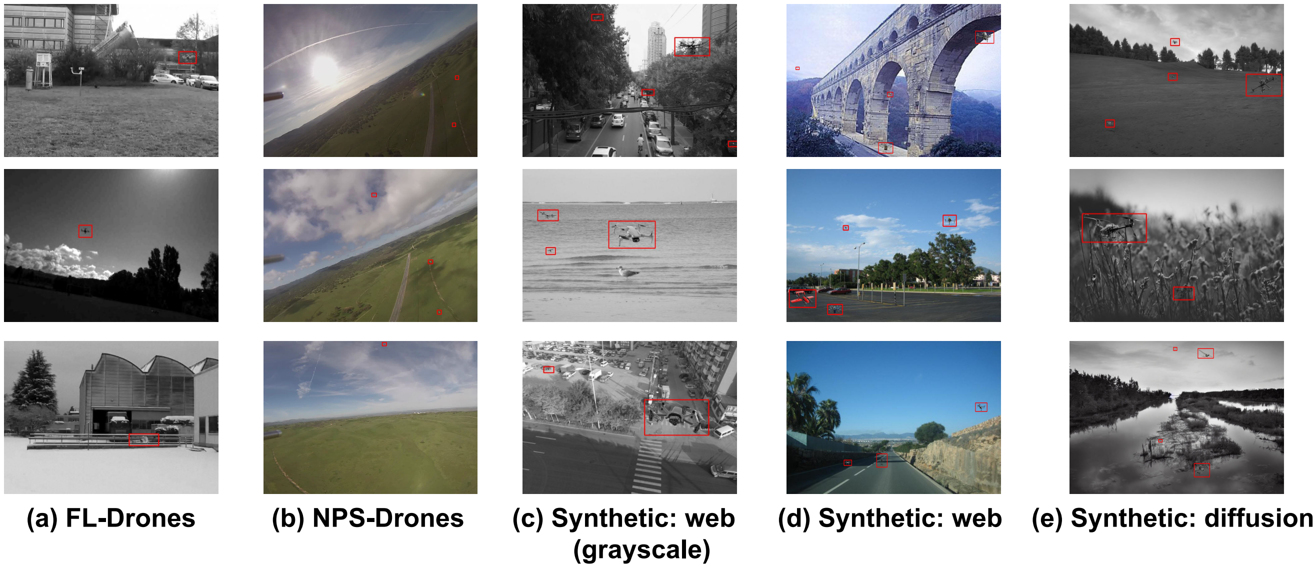

Fig. 1. Sample Training Images from (a) FL-Drones. (b) NPS-Drones. (c) and (d) Synthetic images using backgrounds scraped from the web, are more diverse than the existing datasets. (e) Pre-trained Latent Diffusion models can be guided to generate synthetic backgrounds that are similar to FL-Drones.

Fig. 2. Closeup of drones in synthetic images. A wide variety of drones are used, which is not possible in existing labelled datasets.

II. RELATED WORK

Drone Detection: Prior work on drone detection from airborne cameras have utilized two major labelled datasets: FL Drone [3], NPS [4]. Dogfight [5] proposed a new improved set of annotations for these datasets. Specialized drone detection methods such as Li et. al. [6] utilizing customized neural networks have been proposed. However, with the increasing efficiency of standardized object detection methods such as YOLO, and the increase in the computational capacity of edge nodes such as Nvidia ORIN, such architectures are not necessary to achieve real time performance.

Efficient Object Detection: YOLOv1 [7] introduced the first ever realtime object detector, which was further improved in YOLOv3 [8] and YOLOv5 [9]. YOLOv5 achieves state of the art performance on standard object detection benchmarks such as MS-COCO, while achieving realtime performance. EfficientDet [10] modified standard object detectors with ideas such as Bi-directional Feature Pyramid Network and Compound Scaling to achieve a strong performance-efficiency tradeoff.

Diffusion Models: The earliest deep diffusion models were introduced by Sohl-Dickstein et. al. [11] in 2015. Since then hundreds of works have utilized diffusion models for generating various kinds of high dimensional data. We refer the interested reader to survey [12] of the literature for a complete treatment of the topic, while we discuss a handful of papers most relevant for our work.

Denoising Diffusion Probabilistic Models (DDPMs) [13] slowly corrupt the training data using Gaussian noise and train a denoiser to recover the original data from the corrupted version. A multi-step Markovian noise addition and denoising process allows data to be generated from a randomly initialized noise vector. Dhariwal et. al.[14] demonstrates that DDPMs can be significantly improved through guidance from a pre-trained classifier. Ho et. al [15] proposes training a conditional and unconditional diffusion model simultaneously in order to eliminate classifier guidance. Latent diffusion [16] significantly increases the efficiency and scale of generative diffusion by carrying out the diffusion process in latent space. `stable-diffusion` [17] is a state of the art latent diffusion model trained using web scale data and forms the backbone of our method.

III. METHOD

This section provides an overview of our augmentation based and diffusion based processes for generating synthetic data. We also describe the datasets we used and how we trained and tested our models. We utilize the YOLOv5 family of models for our experiments, for two key reasons: YOLO models provide high detection accuracy with low computational overhead, which makes them suitable for deployment on edge devices like drones. Secondly, the YOLO family of models includes models at 5 different scales, which provides a useful setting to study the impact of our technique on models of different capacities.

A. Generating synthetic images using guided latent diffusion

While synthetic images generated using web data have a lot of diversity and help improve and stabilize training, in certain scenarios it is desirable to obtain synthetic data that's similar to an existing dataset in order to improve performance. For this purpose we adapt a large pre-trained latent diffusion model, stable-diffusion-v1-4. We provide guidance to the standard Image to Image pipeline using conditioning on FL-Drones images, along with a patch based Maximum Mean Discrepancy (MMD) Loss between the generated image and FL-Drones images. Prior works in Generative Adversarial Networks have utilized the MMD loss in order to guide networks to match the moments of the distribution of the generated data to the original data. MMD Loss matches first and all higher order moments of the source data and the target. Our system is illustrated in Figure 3. The encoder takes in an input image to generate a latent representation, which is then corrupted through the diffusion process. Then a denoising U-Net network with cross-attention is applied T times to the corrupted latent to recover a reconstruction of the original. The guidance strength of the diffusion process can be varied to control the diversity of the generated images. The effects of varying guidance strength is illustrated in Figure 4.

B. Generating synthetic data using web images and stable-diffusion

We source two sets of images: First, images of drones with no background. The process of selecting these images involved manual review of about 5 seconds per image. We had a total of 315 drone images, 292 of which were used for training, and 23 were set aside for validation. This split was performed randomly.

To generate drone augmentations we rotated the drone to 5 random angles between -15 and 15 degrees about its center and flipped each drone as well. We then cropped out any borders, this allows us to generate tight bounding boxes around the drones. Finally, we apply a 5x5 box blur to 30 percent of the drones randomly.

We sourced 119 images from aerial videos shot with drones, 5,000 ADE images from [18] selected by [19], and 6471 VisDrone images from [20] to use as our background images. We resize our background image to be one of the resolutions of the FL or NPS datasets based on which dataset we are targeting.

For each background, we randomly select between 0 and 4 drones to be placed on the image. Each drone is resized to fit in the bounds of the expected drone size for a target dataset.

Additionally, we generated 3640 background images using diffusion. Here we took 728 images from the FL dataset which have no drones in them, and use stable diffusion with the prompt 'high-resolution grayscale outdoor photograph' and a guidance strength of 0.5, 0.6, 0.7, 0.8, and 0.9 to generate additional background images. The advantage of using diffusion-generated background images is that they are much closer to the real images found in the FL dataset.

In Figure 1 we can see some sample synthetic images as well as real images from the FL and NPS datasets. In Figure 2 we can see what some of the drones look like. In both images, the drones are larger than they are in our synthetic datasets and are outlined in red to make it easier to see what these datasets could look like. The advantage of diffusion images is evident here, as their resolution and quality are much closer typically to the original FL images than the web images we sourced. Fig 4 shows that the diffusion generated backgrounds look quite similar to the original FL images.

We train on both the synthetic and real target dataset together. We typically (unless otherwise indicated) train for 125 epochs across yolov5n/s/m/l/x. This gives us a wider spread of results to compare against other synthetic data generation techniques we have tried.

C. Datasets

We used the FL drone dataset introduced by Rozantsev et. al. [3]. Additionally, we used the NPS drone dataset introduced by Li et. al.[4], with updated annotations for both datasets from Ashraf et. al.[5].

The FL drone dataset consists of 14 greyscale videos. The NPS drone dataset has 50 RGB videos. The video resolutions as well as the minimum, maximum, and average drone resolutions are shown in Table I.

TABLE I
DETAILS OF EXISTING DATASETS

	FL	**NPS**
# Videos	14	50
Resolutions	640×480, 752×480	1920×1080,1280×760
Drone Dimensions		
Min	9×9	10 × 8
Max	259×197	65 × 21
Mean	25.5×16.4	16.2×11.6
Other Attributes		
Color	Greyscale	RGB
Locations	Indoor + Outdoor	Outdoor only

For the FL drone dataset we used two dataset splits. The first one is proposed by us, we refer to this split as the FL-Sequence split. Here we used all the frames in videos 1, 11, 12, 19, 46, 47, 53, and 56 for training, and all the frames in videos 18, 29, 37, 48, 49, and 55 for testing.

The second dataset split is the one used in [5], we call this split the FL-Temporal split. Here only every 4th frame

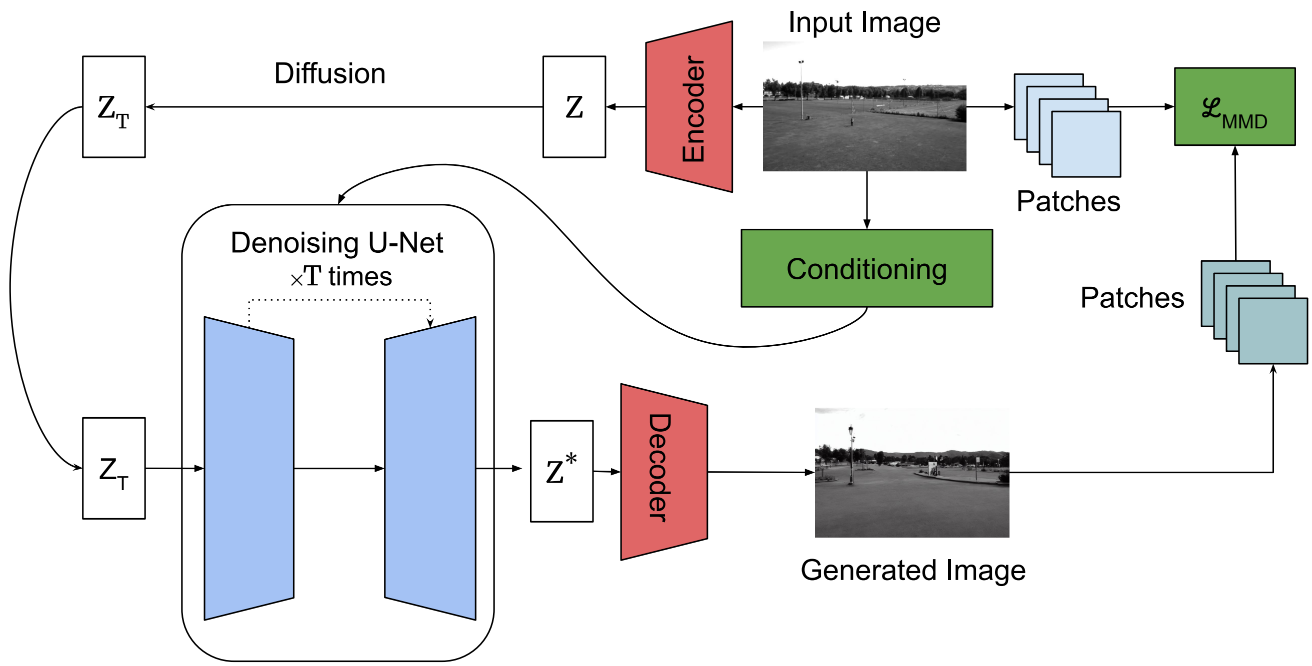

Fig. 3. Overview of the latent diffusion model used for generating Synthetic-Diffusion images. The latent code Z for the input image is computed using the encoder. Then T steps of Gaussian noise addition (diffusion) are carried out to generate the corrupted latent code Z_T. The key part of the network is a de-noising U-Net which is applied T times on the corrupted latent code to recover Z^*, a reconstruction of Z. The decoder generates the output image using Z^*. We use conditioning to guide the U-Net and introduce a patch based Maximum Mean Discrepancy Loss (in green) on the output of the network to match the moments of the generated data distribution to the original dataset.

Fig. 4. (a) Sample background image without drones from FL Drone dataset (b) Three variations generated using diffusion, with progressively weaker guidance strength. The generated images look natural, but with modified details (e.g. grass → crops). Drones have been added to the generated images.

is used. The first half of every video is used for training, and the second half of every video is used for testing.

For the NPS drone dataset, we only use one split which is used by [5]. The first 40 videos are used for training, and the last 10 videos are used for testing.

IV. Results

This section provides an overview of of selected results from our experiments. Our metric for performance is Average Precision @ IoU threshold of 0.5 (henceforth referred to as AP50) and the majority of the experimentation was done with the FL drone dataset, as it was faster to train.

A. Synthetic Data Improves Detection Performance

We apply our optimal synthetic generation strategy to the FL-Temporal dataset split. We tried training over a range of epochs, Table II has the average and standard deviation for training across the range of epochs for the FL and the FL + synthetic data.

As Table II shows, we have between a 3.3% and 6.5%

improvement when we train with synthetic and real data over only real data on average. Synthetic data also reduces run-to-run training variance of the results. Without the synthetic data, due to the smaller size of the FL-Drones dataset, the larger models overfit. As we scale up the models from YOLOv-Nano to XL, the improvement in performance due to synthetic data gets bigger.

TABLE II
EFFECT OF SYNTHETIC DATA ON DETECTION PERFORMANCE
SYNTHETIC DATA IMPROVES THE MEAN RESULT WHILE ALSO REDUCING VARIANCE ACROSS DIFFERENT TRAINING RUNS

	Training Data	
	FL	**FL + Synthetic Data**
Model	**AP50 (in %)** (mean ± std deviation)	
YOLOv5-Nano	65.8 ± 1.0	68.2 ± 0.6
YOLOv5-Small	67.7 ± 1.0	70.0 ± 0.8
YOLOv5-Medium	68.7 ± 0.7	71.3 ± 0.7
YOLOv5-Large	68.1 ± 2.3	72.7 ± 1.2
YOLOv5-XL	66.9 ± 4.7	**73.4 ± 0.1**

B. Finetuning on Real Data

Here we show our results from finetuning the best FL + Synthetic-Diffusion models for 30 epochs on just the FL data. Table III shows that we increase performance with this finetuning on real data.

TABLE III
FINETUNING ON REAL DATA AFTER TRAINING ON SYNTHETIC DATA IMPROVES THE RESULTS BY REDUCING DOMAIN SHIFT

	Synthetic Data
	Diffusion
Training (125 epochs)	73.5
+ Finetuning on FL (30 epochs)	**75.1**

Finetuning helps us achieve our strongest result for the FL independent sequence split. Finetuning models trained with real and synthetic data with only real data is a significant performance boost, as it helps ameliorate domain shifts between real and synthetic data. We can see that this doesn't always boost performance in Figure 5, but it does improve performance for the larger models, such as YOLOv5-L/XL.

C. Applying Synthetic Data to NPS Drones

After determining the best means to generate synthetic data and train drone detection models for the FL-Drones dataset, we apply these ideas on a new dataset, the NPS drone dataset.

After building our initial model, we fine tune it for 10 epochs on real NPS data, and show that it boosts performance in Table IV. The reason we chose yolov5x6 over yolov5x is because the resolution of the NPS dataset is much larger than the FL dataset, and yolov5x6 is better suited for large resolution images. The performance gain from using synthetic data on the NPS dataset is 1.3%.

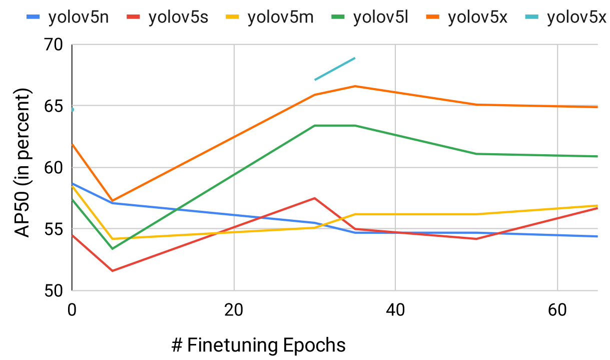

Fig. 5. Effect of number of finetuning epochs.

TABLE IV
SYNTHETIC DATA FOR NPS-DRONES

	Training Data		
Model	**NPS**	**+ Synthetic**	**+ Finetune on NPS**
YOLOv5x6	93.7	94.5	**94.9**

D. Synthetic Data Helps Generalization Across Datasets

Here we compare how well we can generalize and detect drones across datasets.

When we finetune a strong FL synthetic data model for 10 epochs with 10% of the NPS training data and the synthetic data intended for NPS, we achieve a score of 93.4 on the NPS dataset. Additionally when we finetune for 30 epochs a strong NPS + synthetic data model with 10% of the FL training data and the synthetic data intended for FL, we achieve a score of 58.0 on the FL dataset.

This demonstrates that the synthetic data can help create a robust model which can be cheaply pivoted to have reasonable performance on another dataset, with just a fraction of the training data needed to typically achieve this result.

TABLE V
MODELS TRAINED WITH SYNTHETIC DATA GENERALIZE ACROSS DIVERSE DATASETS

	Training Data			
	NPS + Synthetic	**+ Finetuned (10% FL)**	**FL + Synthetic**	**+ Finetuned (10% NPS)**
FL	4.64	**58.0**	73.8	
NPS	94.5		27.6	**93.4**

E. Ablation Studies

This section covers some of the noteworthy ablation experiments that we performed.

1) Synthetic Data Resolution: Here we show the results of matching the background image resolution for the synthetic data to the video resolution of the FL drone dataset. The FL drone video resolution is either 640×480 or 752×480, so we randomly resize the background images to be one of the two. Our results show that matching the resolution is effective in Figure 6.

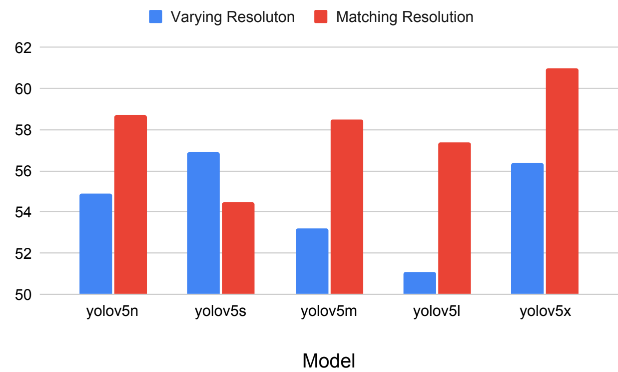

Fig. 6. Effect of matching synthetic image resolution to target dataset.

2) Varying Angles: In this experiment, we varied the angles we used to rotate the drones to generate augmentations. Here we tried 5, 7 or 10 random angles, all within -15 to 15 degrees. We can see in Table 7 that 5 randomly selected angles tend to work best.

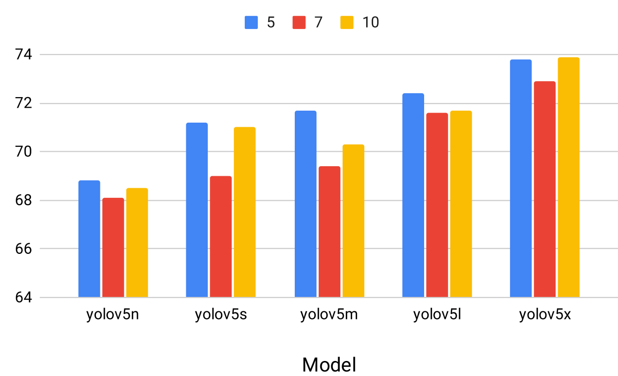

Fig. 7. Effect of number of angles used for drone image augmentation.

F. Comparing to State of the Art

We compare our results with 3 state of the art methods and a simple YOLOv5 baseline. On FL-Drones we provide results on both the previously used FL-Temporal split and our new FL-Sequence split. The FL-Sequence split is a much harder setting since the test and train frames come from different videos of the dataset, whereas in the FL-Temporal split each video is split in half, with the first half used for training.

On the proposed FL-Sequence split our best result (in AP50) is 68.7, which is a significant improvement (**15%**) over Mask R-CNN and YOLOv5 baseline trained without synthetic data.

The current SOTA for the FL-Temporal split and NPS is held by Dogfight [5]. As shown by Table VI, we beat SOTA by 4.3% for the FL dataset and 6.6% for the NPS dataset.

TABLE VI
COMPARISON WITH STATE OF THE ART
RESULTS ARE AVERAGE PRECISION AT IOU THRESHOLD OF 0.5
4.3% GAIN ON FL-DRONES AND **6.6%** GAIN ON NPS-DRONES

	Data		
Model	**FL-Sequence**	**FL-Temporal**	**NPS**
DogFight [5]		72.0	89.0
Mask-RCNN [21]	44.4	68.0	89.0
MEGA [22]		65.0	83.0
YOLOv5 [9]	53.4	71.2	93.7
Ours (YOLOv5+Synthetic)	**68.7**	**75.1**	**94.9**

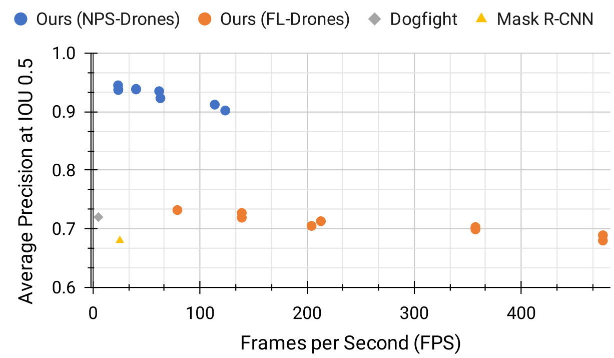

Fig. 8. Performance of our models on FL-Drones and NPS-Drones. Performance of Dogfight and Mask R-CNN on FL-Drones is also indicated. We are able to achieve real time performance and outperform other models.

G. Performance

In order for the drone to successfully avoid collisions with other drones in airborne scenarios, it is important for inference time to be fast. Drones can only carry hardware with a limited amount of computational power. In Figure 8 we graph the frames per second (fps) vs the performance of our models on a single V100 GPU. We can see that for our best model which significantly beats SOTA performance, the fps is approximately 25 and 75 frames per second for NPS and FL respectively. But for slightly lower performance (3% lower for NPS and 6% lower for FL), we are able to achieve approximately 240 fps for NPS and 470 fps for FL. As present generation embedded compute modules like Nvidia Jetson AGX ORIN increasingly match the performance of older datacenter GPUs like the V100, these numbers are a good estimate of the performance in real-world conditions.

V. CONCLUSIONS

In this work we develop a novel method for generating synthetic data by guiding latent diffusion. Our extensive experiments demonstrate that the use of synthetic data significantly improves drone detection on two different datasets. Synthetic data also helps improve hard generalizability across datasets collected in different continents, while lending additional stability to training.

REFERENCES

[1] V. J. Reddi, C. Cheng, D. Kanter, P. Mattson, G. Schmuelling, C.-J. Wu, B. Anderson, M. Breughe, M. Charlebois, W. Chou, R. Chukka, C. Coleman, S. Davis, P. Deng, G. Diamos, J. Duke, D. Fick, J. S. Gardner, I. Hubara, S. Idgunji, T. B. Jablin, J. Jiao, T. S. John, P. Kanwar, D. Lee, J. Liao, A. Lokhmotov, F. Massa, P. Meng, P. Micikevicius, C. Osborne, G. Pekhimenko, A. T. R. Rajan, D. Sequeira, A. Sirasao, F. Sun, H. Tang, M. Thomson, F. Wei, E. Wu, L. Xu, K. Yamada, B. Yu, G. Yuan, A. Zhong, P. Zhang, and Y. Zhou, "Mlperf inference benchmark," in *Proceedings of the ACM/IEEE 47th Annual International Symposium on Computer Architecture*, ser. ISCA '20. IEEE Press, 2020, p. 446–459. [Online]. Available: https://doi.org/10.1109/ISCA45697.2020.00045

[2] "Jetson benchmarks," Aug 2022. [Online]. Available: https://developer.nvidia.com/embedded/jetson-benchmarks

[3] A. Rozantsev, V. Lepetit, and P. Fua, "Flying objects detection from a single moving camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[4] J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, and C. Bouman, "Multi-target detection and tracking from a single camera in unmanned aerial vehicles (uavs)," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4992–4997.

[5] M. W. Ashraf, W. Sultani, and M. Shah, "Dogfight: Detecting drones from drones videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 7067–7076.

[6] J. Li, D. H. Ye, M. Kolsch, J. P. Wachs, and C. A. Bouman, "Fast and robust uav to uav detection and tracking from video," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 3, pp. 1519–1531, 2022.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[8] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[9] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, A. Hogan, lorenzomammana, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, F. Ingham, Frederik, Guilhen, Hatovix, J. Poznanski, J. Fang, L. Yu, changyu98, M. Wang, N. Gupta, O. Akhtar, PetrDvoracek, and P. Rai, "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," Oct. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4154370

[10] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[11] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 2256–2265. [Online]. Available: https://proceedings.mlr.press/v37/sohl-dickstein15.html

[12] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," 2022. [Online]. Available: https://arxiv.org/abs/2209.04747

[13] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[14] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.

[15] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[16] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.

[17] M. Lisa and H. Bot, "Stable Diffusion," 2022. [Online]. Available: https://github.com/CompVis/stable-diffusion

[18] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5122–5130.

[19] A. Bilogur, "Ade20k outdoors," Jan 2020. [Online]. Available: https://www.kaggle.com/datasets/residentmario/ade20k-outdoors

[20] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, October 2021, pp. 2778–2788.

[21] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[22] Y. Chen, Y. Cao, H. Hu, and L. Wang, "Memory enhanced global-local aggregation for video object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 337–10 346.