OnlyPlanes: Incrementally Tuning Synthetic Training Datasets for Satellite Object Detection

Phillip Hale, Luke Tiday, Pedro Urbina
Microsoft

Abstract

This paper addresses the challenge of solely using synthetic data to train and improve computer vision models for detecting airplanes in satellite imagery by iteratively tuning the training dataset. The domain gap for computer vision algorithms remains a continued struggle to generalize when learning only from synthetic images and produce results comparable to models learning from real images. We present OnlyPlanes, a synthetic satellite image training dataset of airplanes in a top-view that contains 12,500 images, 132,967 aircraft instances, with 80 fine grain attributes. We detail the synthetic rendering process to procedurally-generate and render training images and introduce the synthetic learning feedback loop that iteratively finetunes the training dataset to improve model performance. Experiments show the prior performance degradations and false positives when testing the model on real-world images were corrected when incrementally incorporating new features to the training dataset such as diverse plane formations and altitude variations. We demonstrate the promising future for continually improving models by tuning the synthetic training data developed in an iterative fashion to effectively close the synthetic to real domain gap. The OnlyPlanes dataset, source code and trained models are available at https://github.com/naivelogic/OnlyPlanes.

Index Terms- aerial object detection, satellite imagery, synthetic images

1. Introduction

Computer vision methods for geospatial analysis (e.g., satellite image understanding, aerial object detection) continues to be in high demand for researchers, governments and commercial customers providing automated and advanced capabilities such as monitoring changes in large geospatial areas, ecological surveillance, and resource management. This progress is in part driven by advancements in state-of-the-art Deep Neural Networks (DNNs), satellite sensor technology and the availability of publicly annotated aerial image datasets [1] [2] [3]. However, applications in geospatial analysis are limited due to imbalanced occurrences of real world objects, massive variation in image scale, orientation of objects due to the top-down view, diverse perspectives, and inaccurate annotations to enable computer vision models to learn and generalize.

A common approach in computer vision when faced with such challenges is to perform data augmentation, develop novel algorithms or collect more images to label. When it comes to satellite imagery compared to generic object detection datasets such as MSCOCO [4], satellite imagery contains higher resolution with thousands of pixels, resulting in massive variation in scale making objects proportionally smaller with higher overlap between object clusters [5]. Efforts to manually collect and annotate real-world satellite images to train a DNN are challenged with slow, expensive and error prone processes (e.g., inaccurate labels, bias). Therefore, using computer graphics to create synthetic top-view satellite training datasets has demonstrated achievements in various computer vision tasks [1] [5] [6]. Synthetic datasets provide accurate annotations, the ability to incorporate rare occurrences, and enable full control over how objects in an image are simulated (e.g., scale, number of instances, variations introduced). However, the synthetic to real domain gap continues to be difficult to overcome leading to some conclusions that synthetic data alone is difficult to generalize to real-world scenarios.

In general, incorporating synthetic datasets into the DNN training process provides more variability and flexibility to tune the dataset and produces more instances of difficult edge cases to address low performance. Microsoft's Mixed Reality synthetic rendering pipeline was utilized to create a synthetic image dataset for plane detection in aerial images, called OnlyPlanes. OnlyPlanes contains 12,500 images, 132,967 aircraft instances, with 80 fine grain attributes. The OnlyPlanes dataset was constructed incrementally through the synthetic learning feedback loop that we introduce to minimize the domain gap and improve object detection results. The synthetic learning feedback loop is a process that involves stacking multiple incremental synthetic datasets to train and evaluate a model and address areas of improvements by tuning elements of the dataset. Each increment incorporates new targeted features that reduce the synthetic to real domain gap and improve model performance by subsequently addressing previously identified false detections scenarios in real world images.

This paper concentrates on binary plane object detection and instance segmentation tasks by training a Faster-RCNN and a Mask-RCNN only on the OnlyPlanes synthetic dataset. Model accuracy performance was tested on real-world aerial object benchmark datasets to validate the effectiveness of the proposed synthetic learning feedback loop method to improve performance degradations and demonstrate the ability to generalize without learning from real images.

This paper presents the following contributions: In the beginning, we detail the synthetic generation process to create the OnlyPlanes dataset and introduce the synthetic learning feedback loop. We detail the training methodology, experiments, and test the model performance on real-world datasets and benchmark results with similar models. Finally, further analysis is presented

as ablation studies demonstrating the effectiveness of constructing a synthetic dataset in an iterative approach and the ability to target prior false positives that improves overall performance.

2. Related Work

In this section, we focus on reviewing related work on satellite image datasets, computer vision methods for satellite imagery and synthetic to real domain gap strategies.

2.1. Satellite Datasets

Recent advancements in satellite sensors to produce high-quality high-resolution images have led to an active research area in geospatial analysis with more publicly available datasets that specialize in satellite image understanding. Published datasets used in similar research areas for geospatial analytics include [3] [7] [8] [9] [10].

Common challenges faced when utilizing real-world satellite datasets for training models are that many datasets contain imbalanced object instances, massive variation in scale, and rare edge cases that require more examples to produce a generalizable model. One approach to address limitations in datasets is using computer graphic software to generate synthetic datasets. He et al., [5] generated a synthetic dataset generated using the Unity game engine task to detect ships from satellite imagery. The performance when training with this is synthetic training dataset and tested on real world aerial imagery for identifying ships concluded a number of limitations regarding the synthetic and real domain gap, imbalanced in ships locations in different water ways (e.g., harbors, coastlines) and limited variation in ship types and textures [5]. Shermeyer et al., [1] approached object detection similarity by creating a synthetic dataset of aircraft by creating a dataset included annotations for both object detection and instance segmentation and fine-grained attributes for aircraft recognitions. Then a mixture of both synthetic and real image dataset was used to train a detection model. Clement et al. Xu et al. [8] developed a synthetic aerial object detection dataset for zero-shot or few-shot learning and concluded limiting gaps in training only with the synthetic dataset performed poorly without training with real-world images due to low-level gaps (object texture and shading) as well as high-level gaps (e.g., the selection scene content, and its spatial configurations) [6].

2.2. Computer Vision tasks

Object detection in aerial imagery is a fundamental challenge that is actively researched seeking methods to establish data-driven and scalable understanding in the remote sensing domain. The objective of object detection is for a computer vision model to identify an instance of an object given a specified location indicated as a bounding box. Other computer vision tasks for satellite imagery include instant segmentation [11], object oriented [12] [3] [13] and general scene understanding [14].

2.3. Synthetic vs Real Domain Gap

Recently, a number of computer vision tasks utilizing DNNs for object detections have produced methods to improve performance by incorporating synthetic data in the training process. Synthetic image datasets for training computer vision models are artificially produced by computer graphics software [15] or algorithm like a generative adversarial network (GAN) [12] that reflects real-world data.

To maximize the value of the synthetic data for training requires domain adaption where thousands of simulations of the object and the scene incorporates a high amount of variability of features to generalize the DNNs understanding and recognition when it comes to real-world scenarios. Domain adaption aids in addressing the challenge of the domain gap which is the space short of the perfect predictions an AI model would make if it was trained on the exact situation in the real world [16]. To create such exact situations a common method is to create groupings of identical features from the real image to the synthetic image. One study approached this domain gap using a GAN to restyle the synthetic images using style transfer methods such as CycleGAN [17]. To also address the domain adaption challenge using GANs, SimGAN [18] improved synthetic images of airplanes realism by training a discriminator to identify high domain gap instances and a generator that refines the images. These methods depend on the GAN mapping process to map onto a subspace of realistic images to localize and improve synthetic training data [12]. In contrast, the proposed method explores a method to directly target the synthetic to real gap in a more controlled feedback loop where variations can be created through the synthetic rendering process than a GAN like black box.

Other non-GAN related existing methods used to create more realistic synthetic training datasets to address the domain gap includes cropping larger images to remove labeling inconsistencies at the boundaries, manually removing artifacts in the background scenery, adjusting lighting based on perceived differences with real imagery and hand crafting unique image compositions based on use case observed in real imagery. Closer to our task of object detection, Wood et al. demonstrated how to minimize the domain gap at the source by generating highly realistic synthetic data [15].

In general, incorporating synthetic datasets into the DNN training process provides more variability and flexibility to tune the dataset and produces more instances of invariant features that continue to perform low. Extending pervious work [15], we demonstrate the effectiveness of incorporating an iterative training process by incrementally building a synthetic dataset.

3. OnlyPlanes Dataset and Synthetic Framework

In this section we detail the design and methods used to create the OnlyPlanes dataset using the synthetic learning feedback loop. In Section 3.1, we describe the process used for generating synthetic images with Microsoft' Mixed Reality synthetic pipeline. In Section 3.2, we detail the statistical makeup of the OnlyPlanes dataset. In Section 3.3, we introduce the synthetic learning feedback loop used to incrementally construct the OnlyPlanes dataset. Finally, in Section 3.4, we summarize the simulated variability methods used to create robust incremental training datasets.

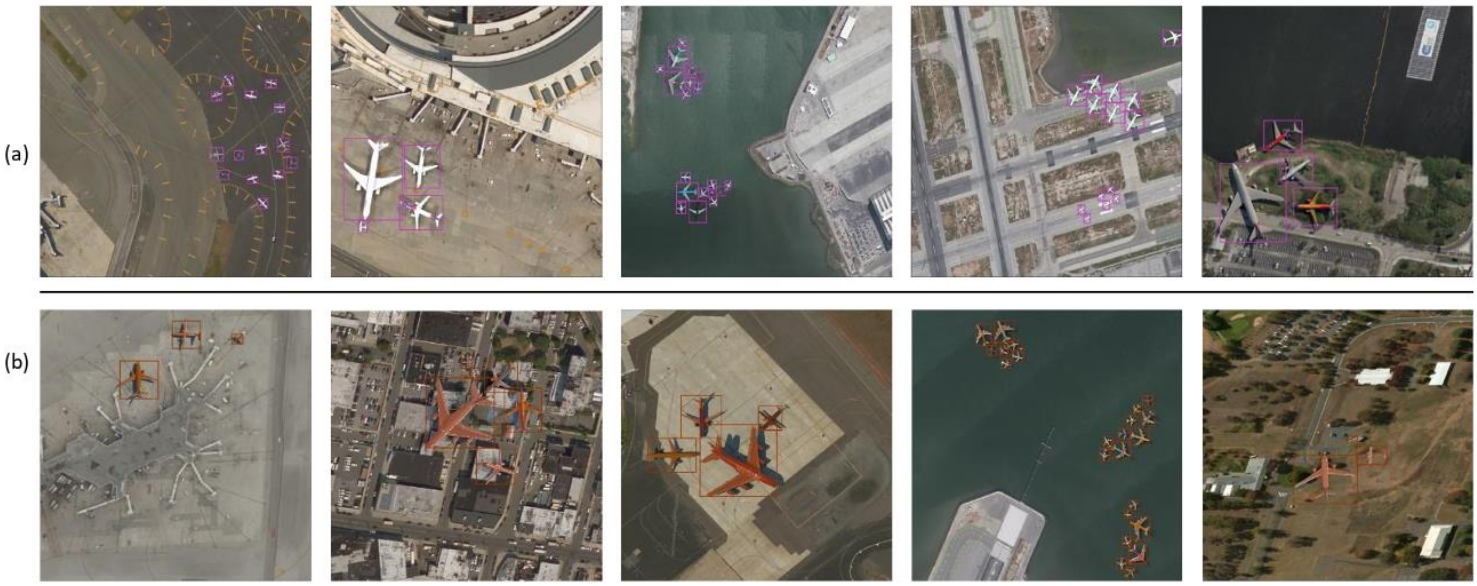

Figure 1. Sample images and annotations from the OnlyPlanes dataset. (a) bounding box annotations and (b) instance segmentation annotations

3.1. OnlyPlanes Synthetic Image Rendering Process

Harnessing the power of synthetics for machine learning begins with an experienced digital artist and computer graphics software that is able to render realistic images procedurally at scale. Microsoft's Mixed Reality synthetic service utilizes these 3D computer graphics and synthetic engine to deliver an annotated dataset of thousands of images and objects to train a computer vision model. The synthetic engine includes an abundance of configurable features to provide near-identical simulated image representation of real-world scenarios. In the case of OnlyPlanes, the synthetic pipeline enabled full control from defining the number of planes to be generated in a specified airport region to augmenting the texture of the plane models to incorporate different colors and branding materials.

The rendering framework presented in

Figure 2, is broken into four components: 1) generate clean airport 2D backgrounds used as the synthetic image backplate, 2) generate 3D plane models, 3) define rendering configurations and parameters (e.g., plane color, lighting conditions); and 4) render a synthetic dataset with annotations in a COCO format that is ready to train a computer vision model.

2D Airport Backplates. To make the synthetic airplane dataset appear realistic, ten 2D real-world, large-scale airports were collected from Bing Maps1. These airports were high resolution that contained tens of thousands of pixels. For example, the Canberra international airport in Australian, featured in Figure 7 (a), had a resolution of 19,200 x 21,248. The airports selected for the OnlyPlanes dataset include three airports from Australia and seven airports from the United States. Each of the 2D airports backplates were manually processed and cleaned removing all real-world aircraft instances in the scene. Airport backgrounds were judgmentally selected in an effort to reduce the domain gap from training with synthetic images by creating an asset library of similar airports that would be representative of real-world scenarios.

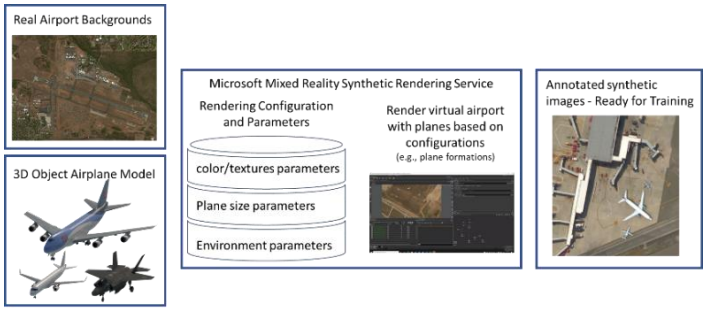

Figure 2. Synthetic rendering process for OnlyPlanes. 3D models procedurally placed on real airport background images (cleaned of planes). Pre-defined rendering configurations and parameters are used to generate large numbers of diverse images that are annotated in a COCO format and ready for training computer vision models.

Generate 3D Plane Models. The second component of the image rendering process is creating an asset library of 3D plane models that represent real airplanes. The 3D plane models were purchased from Hum3d and Turbo-squid websites that included polygonal geometry, materials, textures, and UV maps enabling other customizations to the plane object model. Similarly with the selection of the 2D airport backplates, the curated set of 3D plane assets should be similar to real world plane types to maximize the performance of a model recognizing new aircraft in the wild.

Rendering Settings. To generate OnlyPlanes procedurally in the graphics engine, a 3D camera model was placed at a top-down view to simulate a satellite image perspective with the resolution configured to 1024 x 1024. The 3D camera was configured to generate images of planes randomly within the airport scene. 80 different 3D airplane models were collected and randomly placed in a virtual scene. The rendering pipeline was configured to render a range of specified plane model instances from a single plane in the image up to 20 planes. The default setting used for most of the plane models in OnlyPlanes was to render a random cluster of

[1] Bing Maps API - https://www.microsoft.com/en-us/maps

planes. In Section 3.4 (Simulating Variability) we further detail the rendering methods used regarding plane clustering, plane formations and altitude variation.

Once the planes were rendered, one of ten different 2D airports would be randomly selected as the background to simulate airplanes at an airport. For the final rendering step, two primary lighting sources were used to simulate global and sunlight illumination effects. Global illumination is employed first evenly applying an environmental lighting effect to the entire image. Then sunlight was simulated as a fixed light source across all OnlyPlanes images to give an appearance the images were captured at noon. The simulated sunlight then applied a realistic outdoor lighting effect that is computed by the synthetic engine.

Generating Ground Truth. After the synthetic images have been rendered, the final step is to apply a composite layering function that extracts the final object metadata to create accurate annotations. The metadata is processed by identifying and extracting the bounding box and mask for each plane instance in the image which is saved in the standard COCO [4] annotations file format. At this stage the synthetic dataset is ready to train a computer vision model. In Figure 1 provides examples images and annotations from OnlyPlanes.

3.2. Exploring the OnlyPlanes Dataset

The OnlyPlanes dataset contains 12,500 images and 132,967 instance objects consisting of four categories (plane, jumbo jet, military, helicopter) with 80 fine-grain attributes that define the plane model (e.g., Boeing 737). A single training dataset is provided for both object detection and instance segmentation tasks at 1024x1024 image resolution using ten different airports. The synthetic rendering process provides a unique ability to specify metadata details about each object instance in an image. This dataset can be used for binary plane detection as a single class experiment or utilize the fine-grained class attributes of the plane model for plane recognition.

The distribution of categories shown in Figure 3 (b) illustrates the majority of images contain either a Jumbo Jet which are commercial planes or the plane category which are smaller civilian planes. In Figure 3 (a) shows the number of instances is not proportional to the number of categories, as Jumbo Jet is significantly more represented. While each placement of a plane model was randomly selected during the rendering process, there were 66 different commercial Jumbo Jet models compared to the 7 civilian planes models that were used for the plane category and 7 military planes models. For more details on plane models and attributes refer to Appendix 1: OnlyPlanes Assets.

A common challenging characteristics of satellite imagery dataset is *high spatial resolution where objects in the same type of scene might appear at different scales and orientations* [19]. Figure 4 illustrates the relationship between the width and height of the bounding boxes in the OnlyPlanes dataset as a scatter plot. The color of each point is represented as a heatmap value indicating density of the bounding box size. OnlyPlanes bounding box distribution seen in Figure 4 indicates that most of the bounding box areas contain area dimensions within 30% of the image. Additionally, there exists a smaller secondary cluster of larger sized plane objects that are highly concentrated representing the medium to large size planes.

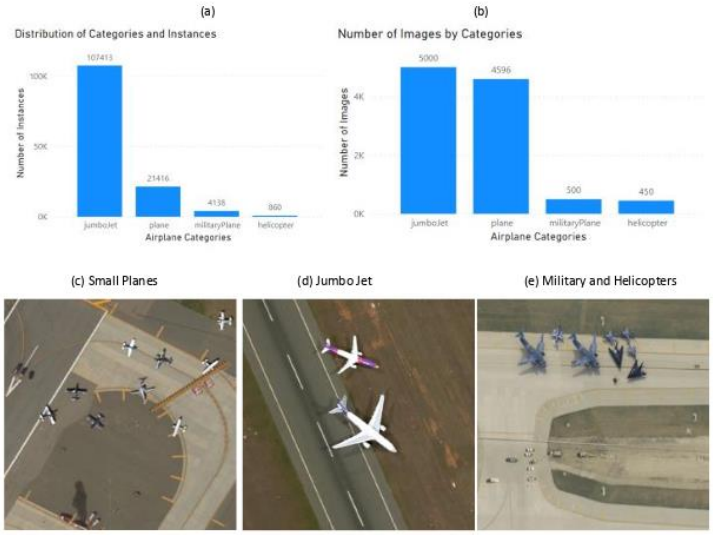

Figure 3. OnlyPlanes dataset statistics and sample images by categories. (a) the number instances by airplane categories, (b) number of images by airplane category. OnlyPlanes. OnlyPlanes contains four class categories: (c) Small Planes, (d) Jumbo Jet and combined is the (e) Military Planes and Helicopters.

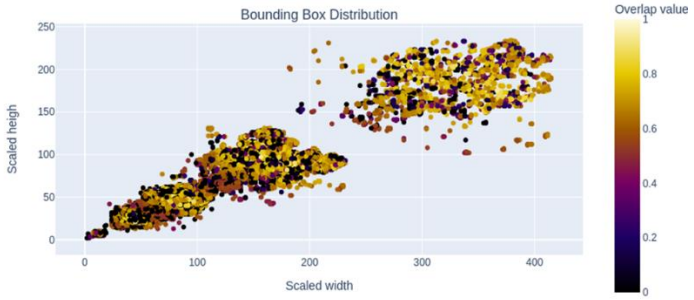

Figure 4. OnlyPlanes bounding box distribution. This figure shows the relationship of the bounding boxes height and width for each object instance in the OnlyPlanes dataset.

3.3. Synthetics Learning Feedback Loop

The OnlyPlanes dataset was constructed incrementally through the synthetic learning feedback loop illustrated in Figure 5. The synthetic learning feedback loop is a process that involves stacking multiple incremental synthetic datasets, to train and evaluate a model and address areas of improvements by tuning elements of the dataset. Each increment incorporates new targeted features that reduce the synthetic to real domain gap and improve the model's performance by subsequently addressing previously identified false detections scenarios in real world images. This continued iterations of reviewing false detections and creating a new incremental dataset to address domain-specific targets enables a repetitive and automated process to achieve improved results by creating an iterative feedback loop between the data designer creating the dataset and the machine learning process.

This incremental process first involves a manual review of model performance using the number of false positives and low inference confidence score as a means of identifying new features

to incorporate into the synthetic training dataset. In the same way,

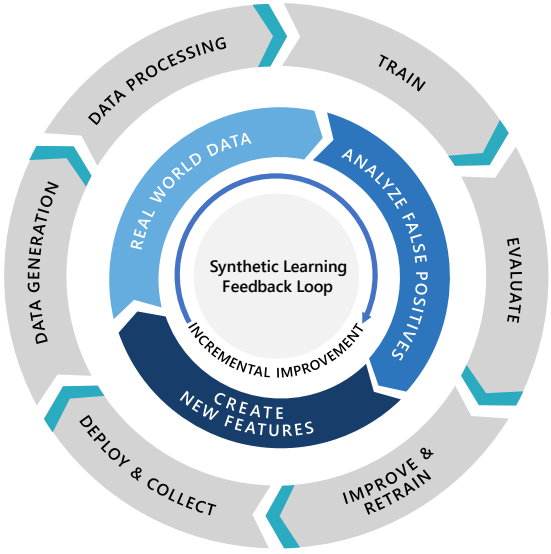

Figure 5. Synthetic Learning Feedback Loop. The basics on incrementally improving computer vision models using synthetic datasets. The synthetic learning feedback loop was used to create a synthetic dataset by incorporating different simulated components of variability to improve prior results.

hyperparameters are optimized for training computer vision models, synthetic training datasets are continuous tuned by incorporating features and rare edge cases incrementally that creates near-identical simulated images that match the poor performing scenarios seen in the wild.

Secondly, the model is trained on the new dataset that incorporated the new targeted features as an incremental component of the overall synthetic training dataset. This overall dataset usually contains the other top performing incremental datasets. Finally, visually inspect how the domain adapted scenarios tested on real-world images to verify improvement of the new features in addition to noting changes in the evaluation metrics (usually mAP). The methods introduced as incremental updates to the OnlyPlanes dataset are detailed in the next Section 3.4 (Simulating Variability) and some results of the correction scenarios are further analyzed in Section 5.5 (Ablation Studies).

3.4. Simulating Variability

Synthetic scenes and objects for training models requires the dataset to incorporate features comparable to those that would be encountered in real-world scenarios with a large number of diverse images and instances. The synthetic learning feedback loop was used to create a synthetic dataset by incorporating different simulated components of variability to improve prior results. The variability introduced in OnlyPlanes was broken down into four tasks: 1) a plane formation behavior called “planes in a row” where planes would be rendered in a linear position as seen in Figure 6 (a), 2) a plane clustering method called “planes at gates” in Figure 6 (b) where planes are parked near airport gates, 3) altitude variation in Figure 6 (c) simulating aircrafts viewed far away and 4) sheering effect in Figure 6 (d) that renders planes similar to an affine transformation.

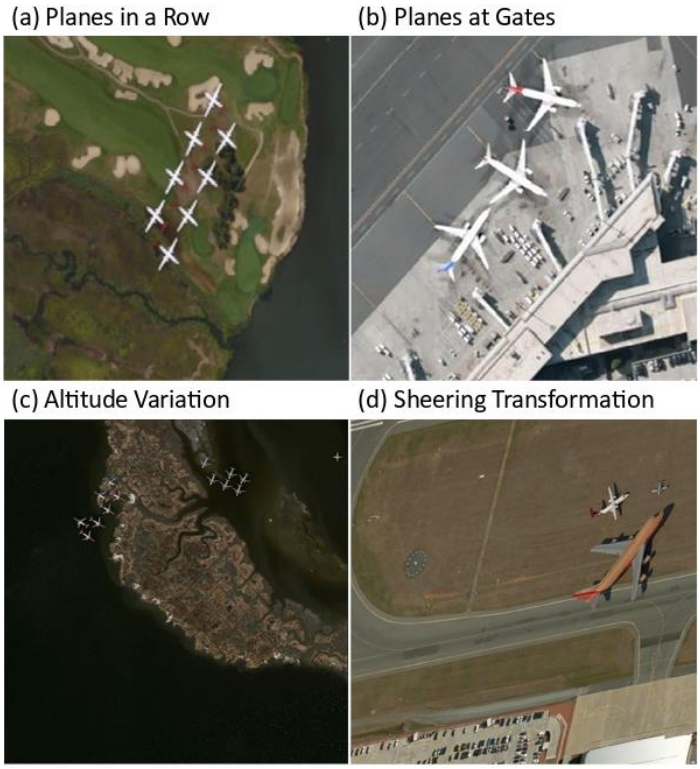

Figure 6. OnlyPlanes sample images for simulating variability tasks. The variability introduced in OnlyPlanes was broken down into four distinct tasks: (a) Planes in a Row, (b) Planes at Gates, (c) Altitude Variation and (d) Sheering Transformation.

Plane Clusters and Fomation Variation. To simulate the various plane clusters and formations of planes, the rendering engine is provided manually crafted annotated landmarks for each of the 2D airport backplates that indicate common plane locations. Random plane clusters Figure 7 (c) are the primary plane formation utilized in the OnlyPlanes dataset. The 3D plane objects were randomly placed in a cluster constrained by the annotated airport-landmark location indicated in Figure 7 (a). For all plane clusters, collision components were set so the rendering engine would automatically remove planes that were placed on top of each other or planes with significant occlusion seen in in Figure 7 (b).

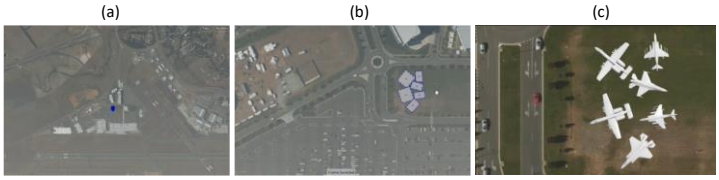

Figure 7. OnlyPlanes plane cluster rendering and airport landmark locations, including: (a) full resolution of 2D airport backplate, (b) annotated airport-landmark locations for airplane collision and occlusion boundary, and (c) examples of rendered airplanes in a random cluster.

Linear Planes Formation. Variability in the formation of the planes incorporated targeted scenarios for how planes would be positioned outside of initial configurations of planes randomly placed in a cluster. One of the target formations was “planes in a row” where planes would be rendered in a linear parking position seen in Figure 6 (a). The defined landmark coordinates for

possible locations of plane instances are illustrated in Figure 8 (a).

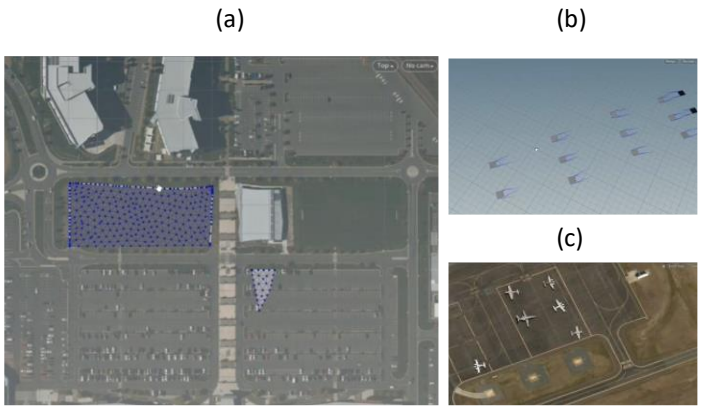

Figure 8. OnlyPlanes formation rendering for planes in a row. (a) the airport landmark areas, (b) planes in a row rendering geometry, and (c) example of rendered planes in a row.

In Figure 8 (b) defines plane orientation and offsets to simulate a linear parking behavior with an example rendered image of planes in a row in Figure 8 (c).

Planes at Gates Formation. In another distinct plane formation, we targeted examples in the domain of planes parked near airport gates and airport hangers. In many real-world airplane datasets, the aircraft usually parked at gates are commercial Jumbo jets, however, prior performance indicated continued false detection of parts of the airport gate as a plane. To address this, an incremental dataset was created to simulate targeted examples of planes near gates seen in Figure 6 (b). In Section 5.5 (Ablation Studies) we further analyze the effects of this incremental dataset and how it improved the model accuracy performance.

Altitude variation. Variability in the camera's viewpoint was incorporated to simulate a higher altitude captured in the scene providing additional examples of planes viewed far away as seen in Figure 6 (c). This variation simulated features to address the massive variation in scale making objects proportionally smaller. This provided an increased scale to create near-identical simulations where more of the airport background is visible on the scenes. This adjustment targets real scenarios, such as in iSAID [8], where full resolution airport images are split and tiled with multiple sizes and planes appear at a distance and smaller.

Sheering Transformation. The final variability task incorporated was simulating a sheering effect on planes as seen in Figure 6 (d). Sheering is defined as the transformation which changes the shape of an object by moving the corners of the image in the fashion of a parallelogram, which is similar to an affine transformation. Plane Sheering effect was incorporated into the dataset to address previous miss detections in the RarePlanes [1] test dataset. This is where a subset of real-world images had scenarios of planes appearing significantly different viewing angles, object shadows and changes in lighting due to the position of the sun relative to the satellite sensor [5]. As part of the incremental feedback loop, a new dataset was incorporated into OnlyPlanes to further target this domain to match the appearance of these real-world scenarios. This required new parameters to be set in the rendering process where the camera model would be placed at extreme focal-length area, so the planes match a similar angled appearance. Additionally, the directional lighting effect was adjusted in the rendering environment setting so the sunlight light source generated identical shadows on the plane to match the real scenarios.

4. OnlyPlanes vs Real-World Datasets

In this section we compare OnlyPlanes with other real-world aerial object detection and instance segmentation datasets. The goal utilizing these datasets is to evaluate the robustness of our dataset when it comes to real-world scenarios presenting more challenges in similar shaped objects, high variation in resolution, lighting, and environmental conditions. As such, following the incremental approach, we concentrated in the areas where the trained network fails to adapt to certain real-world scenarios thus feeding new features to incorporate into the subsequent dataset iterations. Note, some real datasets have multiple categories, for purposes of our evaluations, since OnlyPlanes detector was trained only to perform object detection for airplanes in aerial images, the following datasets evaluated extracted only the binary airplane object categories.

RarePlanes [1] is an aerial image dataset that contains a set of synthetic and real images to demonstrate the value of synthetics data for detecting airplanes. This dataset was selected as the primary evaluation dataset for the OnlyPlanes experiments discussed in Section 5 (Binary Plane Experiments and Results) to demonstrate synthetic learning feedback loop effectiveness to improve model performance for satellite airplane detection. RarePlanes offers a comparative benchmark to compare approaches in training computer vision models with synthetic only, combining synthetic and real and real only trained datasets that benchmark results on the RarePlanes real test dataset.

DIOR [7] is a dataset for object detection in optical remote sensing images. While the dataset contains a large number of object categories, we only utilized airplanes that were extracted for a total of 696 images and 8,042 airplane instances. DIOR provides a unique challenge as this dataset provides a higher dense distribution of plane objects and varying quality of images.

iSAID [8] is a large-scale dataset, for instance segmentation in aerial images that utilizes the same images from DOTA-v1.0 [20] that were collected mainly from Google Earth, the Jilin-1 satellite and the Gaofen-2 satellite. iSAID dataset provides a unique challenge due to high spatial resolution variation, orientation of objects and natural environmental lighting. We used the pre-processed version of the dataset from [11] approach
where images were split into 512 x 512 patches and extreme aspect ratios from the official toolkit were corrected.

NWPU VHR-10 [9] [10] is a high resolution top-view satellite image dataset with ten object classes. As this work was focused on airplane detection, 90 images with 757 airplane instances were selected.

Reported in Table 1 presents a comparison of the aforementioned real-world aerial datasets and our synthetic dataset, OnlyPlanes. OnlyPlanes has the smallest number of images compared to RarePlanes and DIOR. Additionally, outside of the iSAID dataset, OnlyPlanes contains the highest ratio of instance per image with an average of 10.64. iSAID is a unique dataset containing high resolution images emphasizing small objects. The high spatial resolution variation from the iSAID

Datasets	Categories	Annotation	Image Format	Real Instances	Real Images	Synthetic Instances	Synthetic Images	Year Released	Image Width	Instance per Image
RarePlanes [1]	1 to 110	Polygon	PNG	6,812	2710	629,551	50,000	2020	1080	2.51
DIOR [7]	20	HBB	JPG	23,463	192,472	0	0	2019	800	0.12
iSAID [8]	15	Polygon	PNG	655,451	2,806	0	0	2019	800 to 4,000	233.59
NWPU VHR-10 [9]	10	HBB	JPG	3,775	800	0	0	2016	800	4.72
OnlyPlanes (ours)	1 to 80	Polygon	PNG	0	0	132,967	12,500	2022	1024	10.64

Table 1. A comparison of the properties between our OnlyPlanes synthetic dataset and other existing aerial object image datasets.

dataset is a result from images that were collected from various satellite sensors with multiple resolutions. This means the proportion of bounding boxes of the multi-scale instances make the planes sizes larger taking the majority of the image. This is not inherently represented in the OnlyPlanes synthetic training dataset. Data augmentation was implemented to address the scenarios of multi-scale instances gap between the OnlyPlanes dataset and iSAID.

5. Binary Plane Experiments and Results

In this section, we detail the training method used to train DNN models on the OnlyPlanes synthetic dataset and conduct extensive validations on the ability to accurately detect binary airplanes in real-world airplane datasets.

5.1. Training Methodology

The training methodology implemented involved a Faster-RCNN and a Mask-RCNN model solely trained on the OnlyPlanes synthetic dataset. The Faster-RCNN model is used for object detection and the Mask-RCNN model is used for instance segmentation. The selection of the model to train was aligned with the existing RarePlanes [1] binary airplane models to compare performance results. The network architecture implemented for training was the standard two-stage Faster-RCNN and Mask-RCNN models featured with a ResNet-50 backbone and Feature Pyramid Network (FPN) network. Transfer learning was utilized to pre-train the ResNet-50 backbone on ImageNet and the final network was trained on the OnlyPlanes training dataset. Detectron2 was the training platform selected that is built on PyTorch and generally used the default training parameter. Specific customization used in the training process included a learning rate of 0.05, momentum of 0.9, and a weight decay of 0.0001. Additionally, a batch size of 8 with 40,000 iterations was set, a linear warmup period of 600 iterations, the first two stages of the backbone were frozen, and empty annotations were not filtered. All experiments ran in Azure Machine Learning on a single Nvidia V100 GPUs compute cluster.

Data Augmentation. The training process implemented data augmentation using Albumentations to introduce additional noise and variations to the synthetic dataset. A data augmentation strategy is needed to maximize the benefits of training with synthetic images even when the generated dataset incorporates high variability in plane sizes, locations of planes in different airports and diverse plane formations. The augmentation strategy implemented included random vertical and horizontal flipping, random pixel color variation with brightness, contrast, hue saturation, channel shuffle and grayscale. Additionally, image compression augmentations were incorporated with random downscaling that decreased the image quality and posterize transformation that reduced the number of bits for each color channel. To get additional variation in perceived altitude changes and variation in plane scale sizes random crop and rescaling augmentation was implemented to incorporate similar features at training that address the massive scale characteristics that is often contributed to real-world satellite images. The effects and results of experimenting with data augmentation strategies are further detailed in Section 5.5 Ablation Studies.

Evaluation Metrics. The standard COCO evaluation metrics for object detection were used to report the performance of the model. Specifically, the metrics used were the mean average precision (mAP), precision, and recall. During the development of OnlyPlanes in the synthetic learning feedback loop, mAP_{50} was the primary metric to measure accuracy that considers a correct detection of an object when the intersection over union (IoU) between the ground-truth and predicted bounding boxes is ≥ 0.5. However, when testing the confidence threshold used for was 0.1 in alignment with [1].

5.2. OnlyPlanes Detector Results on RarePlanes

As previously noted, OnlyPlanes and RarePlanes share similar approaches in training and testing that enables a comparative benchmark in strategies. RarePlanes is another plane dataset that incorporates experiments using a mixture of synthetic and real training datasets and evaluating the detection performance on a real aerial object detection dataset. We detail in Table 2(a) the comparison results of our OnlyPlanes dataset and the RarePlanes dataset results published in their paper.

Presented in Table 2(a) reports the performance of our trained Faster-RCNN on the RarePlanes real test dataset for binary plane detection. The OnlyPlanes model out-performs the RarePlanes model trained on synthetic images by a mAP_{50} of 4.07 points. Thereby closing the synthetic to real performance gap where the RarePlanes real model (trained only on real images) and RarePlanes Finetune model (trained on synthetic and finetune with 10% of real images) outperform OnlyPlanes by 5.70 and 4.19 mAP_{50} points, respectively. As mentioned earlier in Section 4 (OnlyPlanes vs Real-World Datasets), the RarePlanes synthetic dataset contained 3.6 time more synthetic training images (45,000 RarePlanes compared to 12,500 OnlyPlanes), nearly 500,000 less plane instances (629,551 RarePlanes compared to 45,000 OnlyPlanes) and trained for half the number of training iterations (80,000 RarePlanes compared to 40,000 OnlyPlanes). This method of continually tuning the synthetic dataset indicates a resourceful and effective technique to reduce the gap between training with synthetic and real imagery for object detection.

Network	Real-World Dataset / Training Dataset	RarePlanes Test (a) AP	RarePlanes Test (a) mAP$_{50}$	RarePlanes Test (a) AR	NWPU VHR10 (b) AP	NWPU VHR10 (b) mAP$_{50}$	NWPU VHR10 (b) AR	iSAID (c) AP	iSAID (c) mAP$_{50}$	iSAID (c) AR	DIOR (d) AP	DIOR (d) mAP$_{50}$	DIOR (d) AR
Faster R-CNN	RarePlanes [1] – Synthetic	54.86	87.03	60.67	66.00	97.90	70.90	32.90	56.50	39.00	45.50	76.60	49.90
	RarePlanes [1] – Finetune	69.16	95.29	73.03	63.70	98.60	69.70	34.20	62.20	40.00	46.50	82.30	51.60
	RarePlanes [1] – Real	73.32	96.80	77.16	65.50	98.60	70.90	35.50	64.40	42.20	46.00	79.60	50.80
	OnlyPlanes (ours)	**59.10**	**91.10**	**65.40**	**73.70**	**98.30**	**78.99**	**48.30**	**73.00**	**59.90**	**51.20**	**88.10**	**57.50**
Mask R-CNN	OnlyPlanes (ours)	60.60	92.00	66.90	76.00	98.60	80.60	53.50	77.70	60.40	55.80	91.30	62.10

Table 2. OnlyPlanes and Other Benchmark Dataset for Aircraft Detection mAP comparisons.

Figure 9. Inference results on various real-world aerial plane detection datasets. Results for (a) object detection and (b) instance segmentation from the OnlyPlanes trained model tested on (c) RarePlanes real test dataset, (d) DIOR dataset and (e) iSAID dataset. Results demonstrate object detection and instance segmentation results accurately in various real-world datasets.

5.3. OnlyPlanes Benchmark Results on other Aerial Object Detection Datasets

To verify the performance and to understand how generalizable the OnlyPlanes models are we further utilized other real-world aerial object detection datasets that had airplane categories. OnlyPlanes trained models were benchmarked on DIOR [7], NWPU VHR-10 [9] [10] and iSAID [8]. Table 2 shows the comparison of OnlyPlanes models with the RarePlanes models that were trained on a mix of synthetic and real images to understand the domain gap effects. This benchmark is used to evaluate general accuracy performance, the ability to compare performance with RarePlanes models and to finetune the OnlyPlanes training dataset via synthetic feedback loop. Thus, using different real-world test datasets for evaluation provided additional false positives scenarios to drive what new synthetic

features to incrementally incorporate into the training dataset to further improve the dataset training performance.

The OnlyPlanes models results on NWPU VHR-10, iSAID and DIOR datasets outperform all RarePlanes binary aircraft Faster RCNN models, including the RarePlanes real model that was trained on only real images. This indicates, there are distinct differences between the OnlyPlanes and RarePlanes training datasets as the OnlyPlanes rendering approach incorporates more types of variation and patterns not contained either the real or synthetic RarePlanes training datasets. With higher results across the other benchmark datasets, we use this evaluation as a primary indicator to verify the effectiveness of OnlyPlanes as a suitable training dataset for detecting planes in aerial imagery. This also demonstrates training solely on synthetic data provides comparable results to those trained on real images. Note, RarePlanes model results on these datasets were not reported in the RarePlanes paper. The RarePlanes models were obtained from the GitHub link defined in the RarePlanes paper and implemented the same evaluation methods described in the paper on the other benchmark datasets.

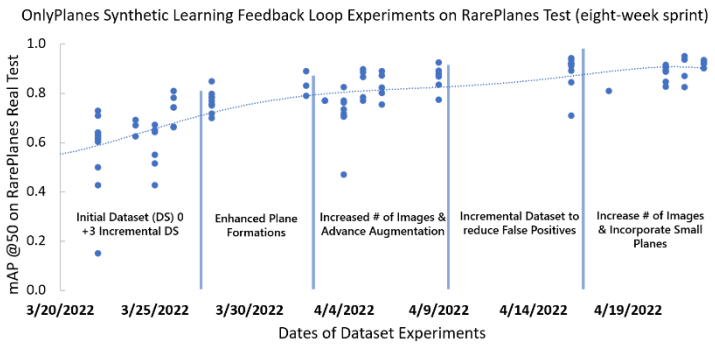

Figure 10. OnlyPlanes Synthetic Learning Feedback Loop Experiment Summary on RarePlanes Test images using the mAP50 scores. Over the course of eight weeks by continually tuning the synthetic training dataset incremental improvements were realized. This demonstrates how continually tuning and incorporating new features into the dataset, not just additional variability and increase in dataset size, but new incremental datasets such as planes at gates, results in the DNN training process to learn more complex representations to improve detection performance.

5.4. Discussion Points

Incremental Improvement. While the main focus of this paper emphasizes the value of incrementally tuning synthetic training dataset, tuning and experimenting with different model hyperparameter settings are also important for improving detection performance. However, as presented in the previous result sections, holding constant the type of network used, the initial resources and time expensed on tuning the training dataset offers substantial gains that often times are overshadowed by the development of a refined network architecture and network hyperparameter optimization. Additionally, we also demonstrate that the incremental approach to building synthetic datasets with the goal of continually adding new features and variations to the training datasets overtime shows a direct increase in performance seen in the dataset.

Figure 10 presents the mAP_{50} scores from the experiments performed to create the OnlyPlanes synthetic dataset tested on the RarePlanes real test dataset. Over the course of eight weeks by continually tuning the synthetic training dataset, incremental improvements were realized. While this synthetic learning feedback loop requires a manual process to review new features to simulate, this is a trade off at the expense of manually collecting and labelling rare scenarios in real-world cases which would presents other limiting constrains in a similar eight week period. By focusing on tuning the datasets and incorporating incremental improvements based on prior performance gaps, the overall detection accuracy improves.

Only Test on Real Images. When it comes to training with synthetic datasets, the evaluation of poor performances are the primary inputs into identifying which features or scenarios needs to be incorporated into the training dataset to improve performance, oppose to finetuning the detection algorithm leaving the training dataset unchanged. In contrast to other methods training with synthetic datasets, model evaluation benefits more when tested on real-world images rather than a holdout validation set of synthetic images. It is common to split a dataset into a training and validation dataset to monitor performance during training to understand if the model is overfitting or if the learning rate needs to be adjusted to further converge, for example. Evaluating performance on synthetic images provides misleading results as the patterns and pixels used to generate synthetic images are easily learned by the model and provides limited value to determine the generalization on real-world imagery.

Data Augmentation Experiments. When training with synthetic datasets we found it as a best practice to think of data augmentations slightly differently for synthetic images when compared to real-image datasets for training. After various experiments we found a simple and effective data augmentation configuration that boosted performance without the need of using Generative Adversarial Network (GANs) to synthesize new surfaces. In contrast to GANs, data augmentation enables more control over the transformations. We implemented data augmentation using Albumentations to introduce additional noise and variations to the generated synthetic dataset. The process did involve some trial-and-error but was shown in Figure 10 and discussed in Section Ablation Study on Data Augmentation the performance significantly improved with augmentation.

5.5. Ablation Studies

This section we present four distinct experiments to understand the effects of the synthetic learning feedback loop used to construct and tune the OnlyPlanes training dataset which was integral in closing the synthetic to real gap and incrementally improving prior model inaccuracies. These training experiments included understanding the impact of no augmentation and pixel-level augmentation, the impact of two key incremental datasets for helicopters and the planes at gates formation. The performance of these training experiments was tested on the RarePlanes [1] real test dataset to benchmark the accuracy for detecting binary airplanes using only a Faster R-CNN network, which is summarized in Table 3.

Ablation studies	mAP	mAP50	AR
No augmentation	38.40	60.20	42.50
Pixel-level augmentation	31.80	57.20	36.00
No helicopters	59.70	90.60	65.40
No Planes @ Gates	59.00	90.60	64.90
OnlyPlanes (ours)	**59.10**	**91.10**	**65.40**

Table 3. OnlyPlanes binary planes ablation studies performance tested on RarePlanes Real test dataset (object detection only).

Ablation Study on Data Augmentation

The experiments performed to evaluate the impact of data augmentation during training included 1) no augmentation, 2) pixel-level augmentation and 3) the final augmentation method indicated as OnlyPlanes (ours) in Table 3. The implemented pixel-level augmentation simply augmented the input image without any modifications to the original bounding boxes (e.g., blur, image compression, brightness). The final augmentation method implemented utilized both pixel-level transformation as well as spatial level transformations that augments both the input image and the bounding boxes (e.g., random flip, random scale, random crop). Training models with just appearance augmentations does not improve detection performance as the mAP_{50} decreased by 3 points compared to no augmentation. The variability of synthetic data is highly important as there needs to be a high level of complexity introduced during training due to the relatively small number of training images to prevent the model from overtraining and limit the bias introduced for scenarios of high similarities in features between training images viz. appearance. Additionally, just augmenting appearance does not address a key challenging characteristic in satellite imagery of massive variation in scale that change the appearance of objects at different scales and orientations. That is why in the final model where data augmentation implements both appearance and spatial level augmentation achieved a mAP_{50} of 91.10.

Investigating the effects of helicopters

The addition of negative objects in the synthetic dataset can aid in developing a robust model. While reviewing false positives from prior experiments, instances of real helicopters were identified as common false detection events (see Figure 11(a)). Following the synthetic learning feedback loop, a new incremental dataset was created to incorporate 860 instances of synthetic helicopters around other civilian and military plane clusters to address this detection gap. For binary airplane detection the helicopter annotations would not be used during training thereby servicing only as synthetic negative object instances. The results presented in Table 3 highlight the detection performance for the model trained on the OnlyPlanes dataset without the incremental helicopters dataset compared to the final model indicated as OnlyPlanes (ours) that incorporated helicopters as negative objects. The final model gains an improvement of 0.50 mAP_{50} points compared to the model without helicopters. While the improvement gain is incremental, Figure 11 (b) highlights three examples where the targeted false detections of helicopters improved and demonstrates the effectiveness and utility of the iteration process when developing synthetic training datasets. Through this investigation we see that the model trained on this new dataset performed better on images with helicopters than those that do not.

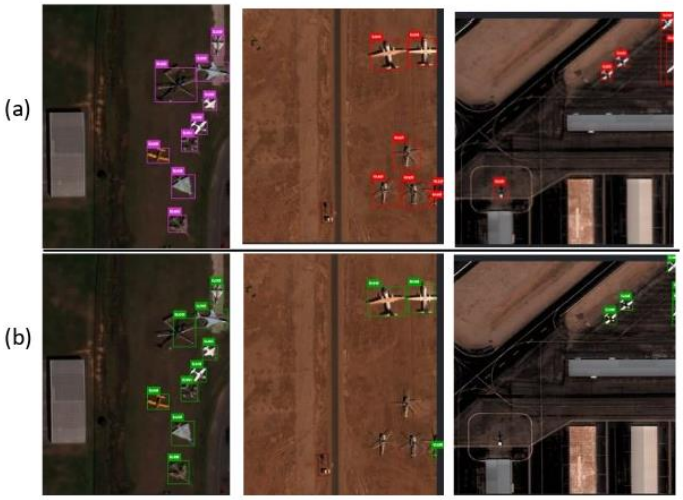

Figure 11. Example on how helicopters as negative objects improved prior false detections. (a) OnlyPlanes without helicopters incremental dataset, (b) final OnlyPlanes

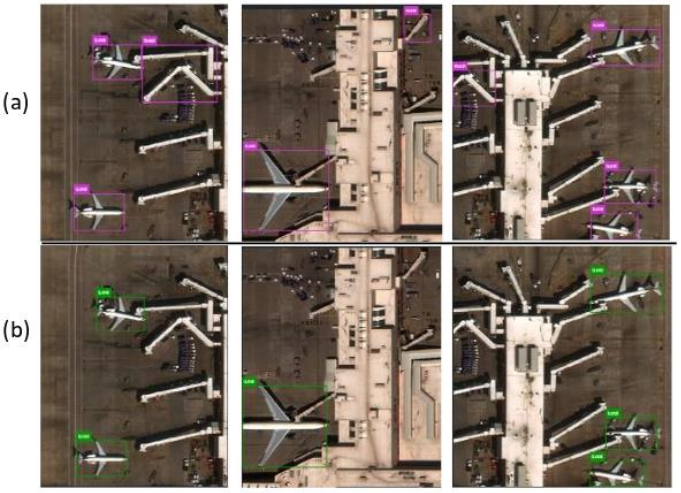

Figure 12. Example on how airport gates false detections were created. (a) OnlyPlanes without "planes at gates" incremental dataset, (b) final OnlyPlanes

Investigating the effects of "Planes at Gates" Incremental Dataset

Presented in Figure 12, provides visual examples of false detections of airport gates in Figure 12 (a) and the error corrections by introducing the incremental dataset of "Planes at Gates" in Figure 12 (b). In this ablation study when training the OnlyPlanes dataset without the planes at gates incremental dataset, the error rate increases, and mAP suffers in Table 3. While this incremental dataset accounted for 4% (500 / 12,500) of the images in the final dataset, benefits can be seen as the false detections of the gates extending from the airport to airplanes is reduced, improving the overall accuracy by 0.5 mAP_{50} points, when the final OnlyPlanes (ours) model is compared to No Planes at Gates in Table 3. This

investigation of the effect of “Planes at Gates” indicates effectiveness on the ability to continually tune the synthetic training dataset by introducing new scenarios to correct prior miss detections. Moreover, even with a small incremental update to the overall dataset, the resulting impact reduces the domain gap between training with synthetic data and the corresponding performance when tested on real images.

6. Conclusion

OnlyPlanes is a synthetic image training dataset for computer vision models to understand airplanes in aerial images. We presented the synthetic framework and detailed the methods for continually iterating and tuning a synthetic dataset with the objective of improving the learning process and performance of computer vision models to detect airplanes. Specifically, we detailed the training approach for using the Faster R-CNN and Mask R-CNN network incorporating FPN, transfer learning, augmentations, and hard negative mining. The results from training with OnlyPlanes were evaluated first on RarePlanes and then to demonstrate the performance to generalize on DIOR, iSAID and NWPU VHR-10 real-world datasets. Across all the real-world benchmark datasets we show the effectiveness of how accuracy for standard computer vision models such as Faster R-CNN can be improved simply by tuning the data than time spent tuning the model, however, dataset tuning to various use cases such as Planes at Gates are easy to configure and scale a massive number of examples, quickly using synthetics than it would be to manually collect new examples and accurately annotate. Implementing this approach to completion, we see no reason for further improvements to even this OnlyPlanes dataset could not reach near perfect accuracy scores with continued dataset iterations to incorporate additional features to be rendered.

References

[1] J. Shermeyer, T. Hossler, A. Etten, D. Hogan, R. Lewis and D. Kim, "RarePlanes: Synthetic Data Takes Flight," pp. 207-217, 01 2021.

[2] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov and B. McCord, "xView: Objects in Context in Overhead Imagery," 2018.

[3] J. Ding, N. Xue, G.-S. Xia, X. Bai, W. Yang, M. Y. Yang, S. Belongie, J. Luo, M. Datcu, M. Pelillo and L. Zhang, "Object Detection in Aerial Images: A Large-Scale Benchmark and Challenges," *IEEE transactions on pattern analysis and machine intelligence,* vol. PP, 10 2021.

[4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. Zitnick, "Microsoft coco: Common objects in context," *European conference on computer vision,* vol. 8693, pp. 740-755, 04 2014.

[5] B. He, X. Li, B. Huang, E. Gu, W. Guo and L. Wu, "UnityShip: A Large-Scale Synthetic Dataset for Ship Recognition in Aerial Images," *Remote Sensing,* vol. 13, p. 4999, 12 2021.

[6] Y. Xu, B. Huang, X. Luo, K. Bradbury and J. Malof, "SIMPL: Generating Synthetic Overhead Imagery to Address Custom Zero-shot and Few-Shot Detection Problems," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 15, pp. 1-1, 01 2022.

[7] K. Li, G. Wan, G. Cheng, L. Meng and J. Han, "Object detection in optical remote sensing images: A survey and a new benchmark," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 159, pp. 296-307, 2020.

[8] S. W. Zamir, A. Arora, A. Gupta, S. Khan, G. Sun, F. S. Khan, F. Zhu, L. Shao, G.-S. Xia and X. Bai, "iSAID: A Large-scale Dataset for Instance Segmentation in Aerial Images," 2019.

[9] G. Cheng, P. Zhou and J. Han, "Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 54, pp. 7405-7415, 12 2016.

[10] G. Cheng, P. Zhou and J. Han, "RIFD-CNN: Rotation-Invariant and Fisher Discriminative Convolutional Neural Networks for Object Detection," 2016.

[11] Y. Liu, H. Li, C. Hu, S. Luo, Y. Luo and C. W. Chen, "Learning to Aggregate Multi-Scale Context for Instance Segmentation in Remote Sensing Images," 2021.

[12] N. Clement, A. Schoen, A. Boedihardjo and A. Jenkins, "Synthetic Data and Hierarchical Object Detection in Overhead Imagery," 2021.

[13] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye and J. Jiao, "Orientation robust object detection in aerial images using deep convolutional neural network," *2015 IEEE International Conference on Image Processing (ICIP),* pp. 3735-3739, 2015.

[14] Y. Long, G.-S. Xia, L. Zhang, G. Cheng and D. Li, "Aerial Scene Parsing: From Tile-level Scene Classification to Pixel-wise Semantic Labeling," 2022.

[15] E. Wood, T. Baltrusaitis, C. Hewitt, S. Dziadzio, M. Johnson, V. Estellers, T. Cashman and J. Shotton, "Fake It Till You Make It: Face analysis in the wild using synthetic data alone," *International Conference on Computer Vision 2021,* 2021.

[16] G. Andrews, "What Is Synthetic Data?," NVIDIA, 8 June 2021. [Online]. Available: https://blogs.nvidia.com/blog/2021/06/08/what-is-synthetic-data/. [Accessed June 2022].

[17] E. Martinson, B. Furlong and A. Gillies, "Training Rare Object Detection in Satellite Imagery with Synthetic GAN Images," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops,* pp. 2769-2776, 2021.

[18] Y. Jiang, T. Zhang, D. Ho, Y. Bai, K. Liu, S. Levine and J. Tan, "SimGAN: Hybrid Simulator Identification for Domain Adaptation via Adversarial Reinforcement Learning," 2021.

[19] G.-S. Xia, W. Yang, J. Delon, Y. Gousseau, H. Sun and H. Maître, "Structural High-resolution Satellite Image Indexing," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives,* vol. 38, 2010.

[20] G.-S. Xia, X. Bai, L. Zhang, S. Belongie, J. Luo, M. Datcu and M. Pelillo, "DOTA: A Large-Scale Dataset for Object Detection in Aerial Images," *CVPR,* 06 2018.

A. SUPPLEMENTARY MATERIALS

A. Appendix 1: OnlyPlanes Assets

Below are the assets utilized to render the OnlyPlanes synthetic dataset for this paper.