Sim2Air - Synthetic Aerial Dataset for UAV Monitoring

Antonella Barisic, Frano Petric, *Member, IEEE*, and Stjepan Bogdan, *Senior Member, IEEE*

***Abstract*—In this letter, we propose a novel approach to generate a synthetic aerial dataset for application in UAV monitoring. We propose to accentuate shape-based object representation by applying texture randomization. A diverse dataset with photorealism in all parameters such as shape, pose, lighting, scale, viewpoint, etc. except for atypical textures is created in a 3D modelling software Blender. Our approach specifically targets two conditions in aerial images where texture of objects is difficult to detect, namely challenging illumination and objects occupying only a small portion of the image. Experimental evaluation of YOLO and Faster R-CNN detectors trained on synthetic data with randomized textures confirmed our approach by increasing the mAP value (17 and 3.7 percentage points for YOLO; 20 and 1.1 percentage points for Faster R-CNN) on two test datasets of real images, both containing UAV-to-UAV images with motion blur. Testing on different domains, we conclude that the more the generalisation ability is put to the test, the more apparent are the advantages of the shape-based representation.**

***Index Terms*—AI-enabled robotics, data sets for robotic vision, aerial systems: perception and autonomy.**

I. Introduction

SYNTHETIC datasets offer a potential solution to data-hungry deep learning models that could drive the development of robotics and bring robotics closer to everyday applications. Today, and even more so in the future, robotic systems such as unmanned aerial vehicles (UAVs) often rely on vision-sensors to sense the world and convolutional neural networks (CNNs) to understand the environment and the objects within it. One such task is UAV monitoring, which involves protecting a specific area of interest from unfriendly UAVs. As the number of UAVs in the world increases every year, both commercial and non-commercial, the UAV monitoring systems are becoming an essential part of all high security areas. The performance of such systems depends on the quality and quantity of available data. To address this issue, researchers are exploring the ability of CNNs to learn from synthetically generated data. Unlike real data, which is expensive, time-consuming, and labour-intensive to collect and annotate, synthetic data can be generated automatically and in unlimited quantities. Synthetic annotations are highly accurate and resistant to human error. Moreover, synthetic data enables balanced datasets that cover all desired versions of real data that may be difficult or impossible to obtain. The synthetically generated data may look indistinguishable to humans compared to real data, but this is not the case with CNNs, and this is called the Sim2Real gap. When we add specific imaging conditions of aerial object detection that affect detection accuracy and robustness, we have a Sim2Air gap to bridge.

The focus of this work is on the creation of synthetic aerial datasets for UAV detection, considering the imaging conditions specific for air-to-air imagery and unconstrained environments. We target two major challenges in aerial object detection, namely long-range detection and detection under changing illumination. The hypothesis of this work is that given the aforementioned challenges, shape-based representation of objects contributes to aerial detection performance. The key contributions of this paper are:

- A shape-based representation achieved by randomly assigned unrealistic textures to improve the performance of aerial object detection in the face of changing illumination conditions in unconstrained environments and the difficult detection of objects that occupy only a small portion of the image, which is very common in aerial object detection;
- A procedural pipeline for generating synthetic aerial dataset for UAV monitoring with diversity of models, backgrounds, lightning, times of day, weather conditions, positions and orientations, camera pan and tilt angles, and distances from camera to object;
- A first publicly available synthetic dataset for object detection of UAVs.

The remainder of the paper is organized as follows: Section II presents recent state-of-art works and findings, while Section III gives a problem description of aerial object detection. Section IV describes the proposed method to bridge the Sim2Air gap, while experimental validation of the proposed hypothesis is described in Section V.

II. Related work

Synthetic datasets: Synthetically generated data has the potential to completely overcome the problem of tedious manual creation of large annotated datasets for training data-driven deep

Manuscript received September 9, 2021; accepted January 11, 2022. Date of publication February 1, 2022; date of current version February 15, 2022. This letter was recommended for publication by Associate Editor T. Patten and Editor M. Vincze upon evaluation of the reviewers' comments. The work of Frano Petric was supported by the European Regional Development Fund under Grant KK.01.1.1.01.0009 (DATACROSS). The work of Antonella Barisic and Stjepan Bogdan was supported in part by European Commission Horizon 2020 Programme through project Twinning coordination action for spreading excellence in Aerial Robotics - AeRoTwin under Grant 810321.

The authors are with the LARICS Laboratory for Robotics and Intelligent Control Systems, Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia (e-mail: antonella.barisic@fer.hr; frano.petric@fer.hr; stjepan.bogdan@fer.hr).

Digital Object Identifier 10.1109/LRA.2022.3147337

learning models. Currently, there are three popular paradigms in the synthetic dataset community: combining real and synthetic data outperforms pure real data, domain adaptation, and domain randomization. The first one has been validated by many researchers [2]–[4], while the other two paradigms continue to be the subject of research. Domain adaptation techniques aim to bridge the Sim2Real gap by minimising the difference between synthetic and real data, while domain randomization techniques aim to randomise synthetic data to the point where the real world is considered just another synthetically generated variant. Tobin *et al.* [5] proposed domain randomization in task of object grasping by randomizing the RGB values of the object texture. A year later, Trembley *et al.* [4] proposed randomizing lighting, pose, and textures in an unrealistic way and adding flying distractors. Both works show compelling performance compared to CNNs trained on purely real data. Another interesting direction proposed in [6] is to explore the texture bias of neural networks. They have shown that texture plays an important role, even more than global object shapes, and that shape-based representations might be more informative. With respect to implementation, there are multiple approaches to generating synthetic data, which can be summarized in two broad categories: 1) deep learning: using models such as Generative Adversarial Networks and Variational Autoencoders [7] to generate synthetic data; 2) 3D rendering: using 3D modelling software such as Unity, Unreal or Blender to generate synthetic data. Two most common approaches in synthetic data generation using 3D rendering are aimed towards using realistic models and/or realistic sensors [8]. Herein, we propose a procedural pipeline that uses realistic models of the object, augmented with unrealistic textures to emphasize geometric shape of the object in the process of learning.

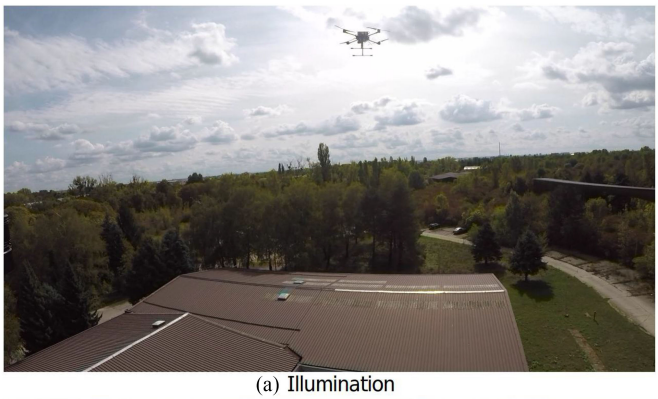
Visual UAV monitoring: Monitoring UAV activity in a predefined area of special interest requires three technologies: detection, interdiction, and evidence collection [9]. The use of image sensors for UAV surveillance, especially for the detection part, has attracted much attention recently. One approach to general UAV-to-UAV detection is to collect and annotate a large amount of real images using different UAV models and types, at different times of the day and under different weather conditions, in different locations to cover a variety of backgrounds, in different positions and orientations, and at different scales and viewpoints. There are several works [10]–[13] that use the above approach to train the detector on real images that had to be collected and manually labelled over a long period of time. In the absence of publicly available datasets, few works [14], [15] have trained UAV detectors on artificial datasets created by pasting object models on background images. Although this solves the data problem, it does not address the specific imaging conditions for drones, does not provide a balanced dataset covering all possible variations of the real world, and suffers from unrealistic lighting due to the simple insertion technique. Therefore, 3D modelling software such as Blender is a more resourceful solution to create synthetic datasets in a controlled manner. Controlling the diversity in a dataset using a procedural pipeline for rendering, like in [16], allows to cover all real-world variations and create balanced datasets. Most similar to our approach is the work of Peng *et al.* [17] who propose a procedural pipeline using physically-based rendering toolkit (PBRT) to render as photorealistic images as possible. However, our approach differs not only in having a larger number of models, a different rendering engine and a lightweight detector, but also in emphasising shape-based representation to reduce texture bias and improve the accuracy and robustness of aerial object detectors.

Fig. 1. A 3D model of a custom quadcopter Eagle, created in Blender [1]. The Eagle on the right has realistic textures that match a real aircraft, while the Eagle in the upper left corner has unrealistic textures.

III. Aerial Object Detection

Object detection from images captured on-board UAV presents additional challenges due to the nature of the aircraft's motion and operating conditions. Aerial perspective means that the detector must be able to detect objects at different pan and tilt angles, at different altitudes, and in 360°. Therefore, it is necessary to distinguish between detection in still images captured by humans, also called generic object detection, and detection on aerial images captured from a moving aircraft. From a robotics perspective, object detection of specific instances such as other UAVs, plants, buildings, etc. is of greater interest.

Intra-class variations, which are the main factor affecting accuracy and challenging the generalisation ability of the detector, can be divided into two categories [18]: intrinsic factors and imaging conditions. Intrinsic factors are variations in appearance within a class, such as different sizes, shapes, textures, etc., which can be addressed by synthetic datasets with high diversity. On the other hand, variations in appearance caused by imaging conditions such as lighting, pose, weather conditions, background clutter, and viewing angles are an innate consequence of unstructured environments. Since UAVs typically operate in such unstructured dynamic environments, aerial imaging conditions must be considered when preparing synthetic datasets. Two problems that are particularly specific to aerial object detection, namely long-range detection and detection under changing illumination, are shown in Fig. 2 using a UAV-Eagle dataset [19] acquired in an unstructured environment. Under varying illumination the same object can appear very different [20], especially in color, depending on the relative position and orientation between the object and the light source in the scene. On the other hand, objects located at a great distance occupy only a small part of the image, which makes them more difficult to detect



(a) Illumination

(b) Small objects

Fig. 2. Imaging conditions of aerial object detection from UAV-Eagle dataset.

because they contain less information about their appearance and require more precise localization. In both problems, shape is more visible and prominent than texture. Therefore, our goal is to guide the object detector towards shape-based detection in order to bridge the Sim2Air gap.

IV. Methodology

In this section, the proposed methodology is described in detail and the main components are outlined.

A. General pipeline

All synthetic datasets in this paper were created in Blender, an open source software for 3D modelling. Blender offers a wide range of tools for modelling all kinds of objects, surfaces, lighting, environments, etc. There are two options available as rendering engines: Eevee and Cycles. The first is for real-time rendering, the second for ultra-realistic ray-trace rendering. In our pipeline, we used Cycles for rendering with 512 samples of path-tracing the light for each pixel. An important feature of Blender is its built-in Python interpreter. User can access Blender objects and tools via Python scripts and manipulate them in a variety of ways. This allows us to take full advantage of the automation and scalability properties of synthetic datasets. For example, our pipeline allows the user to choose how many textures or how many random panning angles to use when rendering a synthetic dataset. Other parameters are scaled accordingly to render the desired number of images. Therefore,

TABLE I
Details of the Datasets Used for Training: **Synthetic Eagle Baseline** (S-Eagle-B), **Synthetic Eagle With Textures** (S-Eagle-T), **Synthetic UAVs With Textures** (S-UAV-T) and **Real UAV** (R-UAV); and for Testing: UAV-Eagle, T_1 and T_2

Train datasets				
	S-Eagle-B	**S-Eagle-T**	**S-UAV-T**	**R-UAV**
Models	1	1	10	$\sim$20
Backgrounds	10	10	10	∞
Textures	1	32	32	N/A
Pitch [°]	[-45, 45]	[-45, 45]	[-45, 45]	N/A
Roll [°]	[-45, 45]	[-45, 45]	[-45, 45]	N/A
Yaw [°]	[0, 360]	[0, 360]	[0, 360]	N/A
Distance $[m]$	[2, 20]	[2, 20]	[2, 20]	N/A
Image No.	32 000	32 000	52 500	11 700

Test datasets			
	UAV-Eagle	**T_1**	**T_2**
Models	1	$\sim$20	1
Backgrounds	3	∞	5
Image No.	510	1300	285

scalability of the dataset is easily feasible with the proposed procedural pipeline. In parallel with image rendering, all objects in the current image are automatically and precisely labelled with bounding boxes enclosing each object. The labels are stored in a appropriate format for later training of the detector.

Considering all the aspects of aerial object detection mentioned in Section III, and to account for the diversity of the real world, we implement variations in the following components within the procedural pipeline:

- 3D models of different quadcopters and hexacopters,
- number of objects in a single image,
- environment maps with different lighting conditions such as daylight, partly cloudy, twilight, etc.
- distance between camera and objects, i.e. scale of objects,
- angle of the camera (pan, tilt and yaw) with respect to the world origin,
- location of the camera with respect to the world origin,
- a mixture of atypical textures.

Each model is rendered in all environment maps from the set, and each texture from a set of mixed textures is assigned to the model. Then for each iteration, a predefined number of random values for pan, tilt, and the distance between the camera and the objects are selected. With such configuration in a scene, a set of images is rendered by animating the yaw angle of the camera around objects. All components except texture are modelled to reflect the real world as closely as possible. Details of the parameters of each dataset used to train the aerial object detector are given in the Table I, where the first letter of the dataset name indicates the type of data (synthetic or real), the middle part indicates the type of models used (Eagle quadcopter only or multiple UAV models), and the last letter indicates the type of technique, if any, used to create the dataset (e.g., -T for texture randomization).

B. Models

The first model in this paper is the Eagle quadcopter. The Eagle is a custom aerial platform that is used for a variety of tasks as its modular design allows for different sensor configurations.

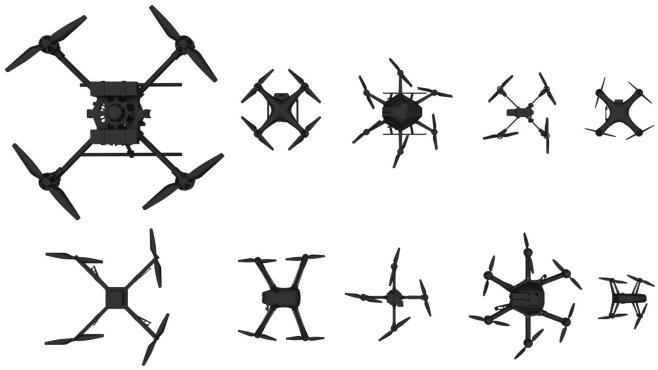


Fig. 3. All 3D models of UAVs used to create a synthetic dataset for aerial object detection.

Fig. 4. Sample of textures used for a texture-randomization technique.

A custom UAV presents a more difficult problem than other commercial UAVs because data (images) collection is more challenging. One should collect a large number of air-to-air images to realistically reflect the imaging conditions of UAV detection from the air. To overcome this problem, we propose to use 3D models of custom aerial platforms such as Eagle. The photorealistic model of Eagle shown in Fig. 1, which closely resembles an actual aircraft, was created using standard modelling techniques in Blender.

In addition to the 3D model of Eagle, another 9 models of quadcopters and hexacopters were used to create a synthetic aerial dataset for UAV monitoring. Three models were found online [21], while the other six models were obtained from Gazebo simulations, mainly from the RotorS [22] library. All these together result in a set of 10 different UAV models, which are shown in Fig. 3.

C. Textures

Each time the generation of a synthetic dataset is started, a different mixture of textures is created (see Fig. 4), which can later be assigned to objects in a scene. Some of the textures are procedurally created by an algorithm that combines different material properties with random colors. Different material properties are implemented by so-called shaders, which determine how light is scattered across the surface of the object. For a predefined number of procedurally generated textures, a shader is selected from four shaders, namely diffuse, glossy, glass and translucent bidirectional scattering distribution function (BSDF). Each shader implements a different mathematical function that describes the reflection, refraction, and absorption of light. A diffuse shader is the most realistic material for UAVs, while the others contribute to the unrealism of the textures. To demonstrate the irrelevance of colors to the detector, each material is assigned a random RGB value for color. The second part of the texture mixture is created by importing image files with textures that are atypical for UAVs, e.g. bathroom tiles, lollipop texture, Christmas ornaments, kitchen towel, wood, chip texture, marble, metal, etc. In the end, a realistic texture of carbon-made UAV is added to the mixture in order to treat the real texture as another random variation. The thus created mixture of textures using both monochromatic and patterned textures, typical and atypical colors, artificial and man-made textures, reflective and non-reflective materials, etc. leads the detector to pay more attention to the shape of the UAV.

D. Environment maps

Instead of modelling a series of complex scenes, a set of environment maps is used to replicate typical unstructured environments of UAVs. The environment maps are implemented using High Dynamic Range Imaging (HDRI), a 360-° panoramic image with extensive brightness data. The HDRI maps not only serve as a background for the scene, but also provide realistic illumination of the scene. This is especially useful in the case of custom aerial platforms, as images containing them are difficult to collect. Collecting images could take months to cover different weather conditions such as sunny, cloudy, foggy, etc., take place at different times of the day to cover all lighting conditions, and would also require at least two UAV operators. All of these problems are addressed with a set of 10 HDRI maps, acquired from Polyhaven [23], with different lighting conditions and different types of outdoor environments.

V. Results

To verify our claims, a general object detector suitable for on-board UAV processing was trained on several datasets. The experiments are divided into two parts. First, we train the detector on datasets based on a single custom model of UAV, with and without applying random unrealistic textures, to show the impact of the proposed technique on the generalisation and robustness of the aerial object detector. Second, we create a larger dataset with different UAV models and compare the performance between synthetic and real data in the UAV monitoring task. The quantitative and qualitative results of how our method improves the detection of objects in the air are presented below. The synthetic datasets are available at https://github.com/larics/synthetic-UAV and the video of our approach can be found at https://www.youtube.com/watch?v=7pPGEk8t_Tw.

A. Training of object detector

In this paper, two object detectors, each representative of its category, are trained and evaluated for the task of aerial object detection. The first one comes from a well-established family of one-stage detectors called You Only Look Once (YOLO) [24], [25]. YOLO detectors treat object detection as a regression

Fig. 5. Examples of images in the synthetically generated datasets S-Eagle-T (top row) and S-UAV-T (bottom row). Both datasets contain randomized textures, while S-UAV-T contains 9 other UAV models in addition to Eagle quadcopter.

problem and use features from the entire image to detect objects, implicitly including contextual information. There are two advantages of YOLO that led us to choose it as an object detector. The first important advantage is the low inference time, which accommodates the limited computational resources of on-board processing. The second advantage is the accessibility of the network implementation along with the continuous development and wide community support, which allows easy portability to any application. In selecting the detector, we looked for a lightweight network and therefore selected a lightweight version, called Tiny, of the fourth generation of YOLO networks. Taking into account that bird's eye objects usually occupy only a small part of the image, the network architecture was slightly modified by adding another YOLO layer to better detect both small and large objects. On the other hand, we chose the Faster R-CNN [26], a representative of the two-stage object detectors, as the second detector to show that the proposed method does not depend on the type of detector. We utilize a base model with a backbone combination of ResNet-50 and Feature Pyramid Network with a $3x$ learning rate scheduler. In general, the Faster R-CNN is a more accurate detector than YOLO, but it requires significantly more time for inference and is therefore generally not suitable for on-board processing.

All training and dataset rendering was performed on a computer equipped with a Intel Core i7-10700 CPU @ 2.9 GHz x 16, an Nvidia GeForce RTX 3090 24 GB GPU, and 64 GB RAM. The YOLO detector was trained within Darknet framework and Faster R-CNN within Detectron2 [27] framework, both using default anchors and starting from weights pretrained on the COCO dataset [28]. For training, we used an image size of 608, a learning rate of 0.00261, a momentum of 0.9 and a decay of 0.0005. Each model was trained for 200 epochs, using a batch size of 64 for YOLO and a batch size of 16 for Faster R-CNN. The best weights were determined using an early stopping method based on the mean average precision (mAP) to avoid overfitting. The training parameters were the same for all datasets.

B. Texture-invariant dataset of Eagle UAV

To test our hypothesis that a shape-based, i.e. texture-invariant, representation of UAVs boosts performance of aerial object detectors, we created two datasets: **S**ynthetic **Eagle** **B**aseline (S-Eagle-B) and **S**ynthetic **Eagle** with **T**extures (S-Eagle-T) dataset. Both datasets were created using a single UAV model, the custom aerial platform Eagle, and contain the same number of images. The main difference is that the baseline dataset was created with a photorealistic texture that mimics the real aircraft: a carbon frame with a synthetic plastic texture for smaller parts of the model, as shown in the lower right corner in Fig. 1. The S-Eagle-T dataset was created with 32 unrealistic textures to accentuate the geometric shape of the UAV. Samples from the S-Eagle-T dataset are shown in the top row of Fig. 5.

The detectors trained on S-Eagle-B and S-Eagle-T datasets were evaluated on three different test datasets, all consisting of real images only. The details are given in Table I. The first dataset is UAV-Eagle [19], which contains images of real Eagle UAV in an unconstrained environment with pronounced illumination effects, distant objects, and a highly cluttered background. The second test dataset is part of our previous work [11], [19], where we collected a large number of real images from the Internet. The labelling was done partly by manual labelling and partly by pseudo-labelling. This dataset consists of 13 000 images which are divided into two parts: a training dataset called R-UAV

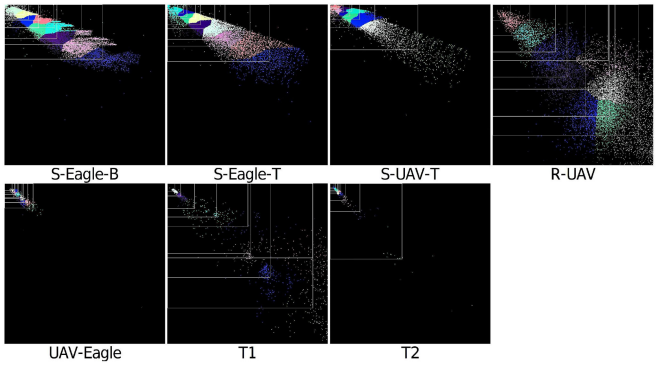


Fig. 6. Distribution of data in the training and testing datasets obtained by the k-means++ algorithm. The colors indicate the 9 clusters of computed anchor boxes that best correspond to the actual size of the objects in the dataset. The coordinates of each pixel correspond to the width and height of each bounding box.

TABLE II
EVALUATION RESULTS OF DETECTORS YOLOV4 TINY AND FASTER R-CNN TRAINED ON BASELINE AND PROPOSED APPROACH

Detector	Train dataset	UAV-Eagle	T_1	T_2
YOLO	S-Eagle-B	**86.84%**	30.11%	64.76%
	S-Eagle-T	85.65%	**47.16%**	**68.47%**
Faster R-CNN	S-Eagle-B	**92.28%**	65.50%	64.68%
	S-Eagle-T	89.01%	**85.39%**	**65.75%**

containing 90% of the images, and a test dataset T_1 containing the remaining 10%. Some of the images in T_1 were taken in the studio, some indoors, some outdoors, and most of the images feature commercial drones. The third and last dataset for evaluation, called T_2, is a test dataset from [10]. The T_2 dataset consists of real images of the author's custom UAV taken in different unstructured environments. The datasets have different domains, from the style of the images to the type of objects and the size of the objects in the images. The T_2 dataset is the most difficult in terms of detecting small objects, as can be seen in Fig. 6, followed by UAV-Eagle. An additional challenge in the UAV-Eagle and T_2 datasets are images of UAVs captured from UAVs, which introduce significant amount of motion blur. The T_1 and R-UAV datasets, on the other hand, contain larger objects.

The evaluation results in terms of the mean Average Precision (mAP) with an Intersection over Union (IoU) threshold of 0.5 are presented in Table II. On the UAV-Eagle dataset, the detectors trained with realistic texture achieve a better mAP value than the detector trained with random atypical textures. This is expected since the synthetic model and texture fit the test dataset almost perfectly. In the test dataset T_1, however, the detectors trained with a shape-based representation outperform the baseline by a large margin, by 17 percentage points in mAP for YOLO and 20 percentage points for Faster R-CNN. The significant difference indicates that by guiding detector towards shape-based representation, we have improved the performance of the aerial object detector, more specifically its generalisation ability and robustness. The T_1 dataset is very different in style from the synthetic datasets on which the detectors were trained. For this reason, the baseline detector with only one texture experiences a large texture-bias that degrades its generalisation ability, while

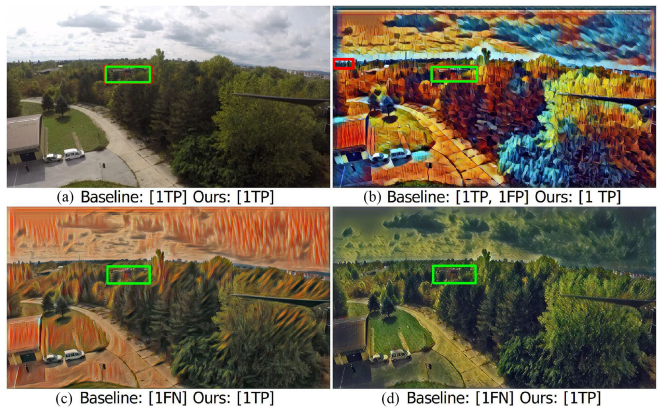


Fig. 7. Detection results of the baseline S-Eagle-B (red) and our method S-Eagle-T (green) on images with different textures. While both detectors accurately detect the original image (a), the proposed method performs better than the baseline method on stylized images (b,c,d) without texture cues. TP stands for true positive, FP for false positive and FN for false negative detection.

the S-Eagle-T detector shows improved accuracy on previously unseen UAVs. To confirm that the detector trained on S-Eagle-T gives more importance to the shape of the UAV, e.g., the frame, propellers, and landing gear, we perform additional tests with YOLO detector. We eliminate texture cues while preserving the shape of the object from the image using a neural style transfer algorithm [29]. While the S-Eagle-B detector struggles without texture cues, the qualitative results in Fig. 7 show that the S-Eagle-T detector handles texture variations well.

In the third evaluation, the S-Eagle-B and S-Eagle-T detectors are compared on the test dataset T_2, which contains images of a custom UAV that were not in the training data of any detector. Using a texture-invariant approach, an increase in mAP of 3.71 percentage points for YOLO and 1.07 for Faster R-CNN is observed. This is further confirmation that our approach improves the performance of the aerial object detector, as it has better generalisation ability and robustness to small objects on T_2 dataset. The larger increase in mAP when evaluating on the T_1 dataset can be explained by the larger difference in domain between the T_1 and synthetic datasets. Thus, the more the ability to generalise is put to the test, the more evident becomes the influence of shape-based representation on the improvement of accuracy.

C. Synthetic vs. Real

For application in UAV monitoring, we have created a synthetic dataset of 52 500 images with 10 different UAV models using the proposed procedural pipeline and texture randomization technique. We named this dataset **S**ynthetic **UAV**s with **T**extures (S-UAV-T). Sample images are shown in the bottom row of Fig. 5. For the following experiments, the YOLO detector is utilized since it is suitable for on-board UAV processing. Following the procedure described in subsection V-A, the detector is trained on S-UAV-T synthetic dataset and the results are presented in Table III. Although we expected to obtain better results with more models, the evaluation on test datasets with real images showed the opposite. Overall, the detector trained on

TABLE III
EVALUATION RESULTS OF THE YOLO DETECTOR TRAINED ON SYNTHETIC DATA, TRAINED ON REAL DATA, AND TRAINED ON SYNTHETIC DATA AND THEN FINE-TUNED ON REAL DATA

Train dataset	UAV-Eagle	T_1	T_2
S-UAV-T	73.80%	42.41%	68.83%
S-UAV-T_f	95.24%	**93.86** %	71.54%
S-Eagle-T_f	**95.90%**	91.23%	**76.44%**
R-UAV	90.38%	91.31%	65.73%

TABLE IV
ROBUSTNESS AGAINST CHALLENGING ILLUMINATION CONDITIONS. COMPARISON OF THE BASELINE AND THE PROPOSED METHOD ON MODIFIED T2 AND UAV-EAGLE TEST DATASETS

Test dataset	Method	S-Eagle-B	S-Eagle-T
UAV-Eagle	Original	**86.84**	85.65
	Challenging Illumination	72.35	**76.95**
T_2	Original	64.76	**68.47**
	Challenging Illumination	46.02	**62.28**

S-Eagle-T outperforms the one trained on S-UAV-T, suggesting that a carefully selected model is sufficient. Other factors that could account for this result are the higher prevalence of small objects in the S-UAV-T dataset and overfitting to synthetic data.

Just as we addressed the gap between aerial and generic object detection in the previous section, we now apply the fine-tuning technique to close the gap between synthetic and real world. The detectors trained on the S-Eagle-T and S-UAV-T datasets are fine-tuned for 20 epochs on real images from the R-UAV dataset with decreased learning rate of 0.00161. To allow a full comparison of the performance of the proposed approach, YOLO detector was also trained on the R-UAV dataset using only real images. The detection results of the trained detectors are shown in Table III. The fine-tuned datasets, indexed by f, show comparable performance. The S-Eagle-T_f detector performs better on datasets whose domain is custom-made quadcopters, while S-UAV-T_f detector performs better on T_1 dataset that contains more UAV models. Moreover, the results show that both detectors trained on a balanced synthetic datasets, built specifically to include diverse poses of the object under different lighting and backdrop, and fine-tuned on real data, perform better or equally well as the detector trained purely on a collection of images obtained and annotated through tedious work.

Apart from the fact that the combination of synthetic and real data is better for all three test datasets, the biggest difference is seen in the T_2 dataset, where S-Eagle-T_f outperforms the R-UAV detector by almost 11 percentage points. We conclude that there are two reasons for this. First, as seen in Fig. 6, the T_2 dataset is most difficult in terms of small object detection which is why our texture-invariant detector performs drastically better. Second, just like UAV-Eagle, the T_2 dataset contains only one type of custom UAV, which are not included in the R-UAV dataset and differ in appearance from the commercial and popular UAVs that make up the bulk of the R-UAV dataset. Therefore, the generalisation capability of detector trained on R-UAV is limited. We strongly believe that these results confirm that introduction of texture-randomization through synthetic datasets places higher importance on the shape of the UAV, which in turn yields improved detection results.

D. Discussion

In unstructured environments, lighting conditions are uncontrollable and unknown in advance. Due to the movement of the target and the camera, varying illumination occurs during UAV monitoring. Our approach not only improves the overall performance of aerial object detection, but also improves the detection under varying illumination conditions. To prove this claim, we perform additional tests. We programmatically simulate difficult illumination conditions on images from test datasets. The quantitative results are presented in Table IV.

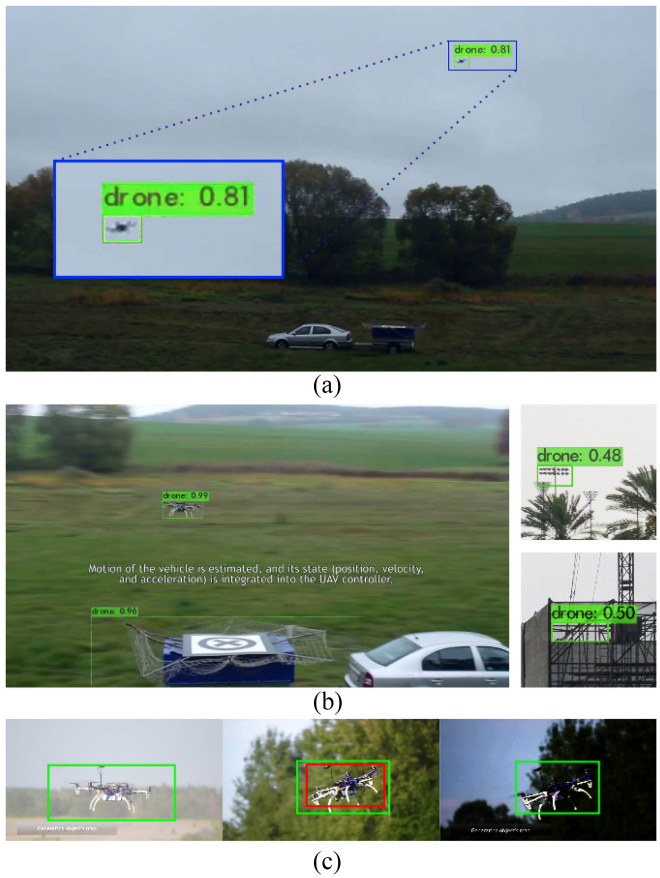


Fig. 8. Detection results on images from the T_2 test dataset of the detector trained on purely synthetic data with unrealistic textures (green). (a) Example of successful detection of distant object where the detector trained purely on real images fails. (b) Example images of false positive detections caused by similarly shaped objects. (c) Detection results of the S-Eagle-B (red) and S-Eagle-T (green) detectors on a bright overexposed image, a normally illuminated image, and a dark underexposed image.

On both datasets, the proposed method of texture-invariant object detectors improves the detection under changing illumination by 16 percentage points in mAP on T_2 dataset and 4 percentage points on UAV-Eagle dataset. The qualitative results can be seen in Fig. 8 c. The synthetic texture-invariant detector (green bounding box) detects the target object under all conditions, in the bright overexposed image, in the normal illumination image, and in the underexposed dark image. The detector without our enhancement is only able to detect the

object in a normally illuminated image. Improvements are also seen in the detection of distant objects, as can be seen in Fig. 8(a), where the S-Eagle-T detector successfully detects the object, while the detector trained on R-UAV fails. The presented results show that the Sim2Air gap can be bridged with the proposed approach and that the accuracy of the real data can even be surpassed. On the other hand, since the detector trained on S-Eagle-T is more attentive to shape, there are situations in which similarly shaped objects lead the detector to false positive detection, as shown in Fig. 8 b. This problem could be solved by including negative samples of shape-like objects in the training dataset. In general, more attention will be paid in the future to reduce false positive detections by using techniques such as flying distractors [4].

VI. Conclusion

In this paper, a texture-invariant object representation for aerial object detection is presented and evaluated on several test datasets with real images. A procedural pipeline for generating synthetic datasets is developed, implementing the proposed technique of randomly assigning atypical textures to UAV models. The results of the evaluation of the synthetically generated datasets confirm that shape plays a greater role in aerial object detection. This is due to the imaging conditions of the aerial perspective in unstructured dynamic environments, where the texture of the object is difficult to discern. Quantitative and qualitative results show that the proposed approach outperforms baseline and real-world data in situations with difficult lighting and distant objects.

References

[1] R. Hess, *Blender Foundations: The Essential Guide to Learning Blender 2.6*. Waltham, MA, USA: Focal Press, 2010.

[2] F. Nowruzi, P. Kapoor, D. Kolhatkar, F. A. Hassanat, R. Laganière, and J. Rebut, "How much real data do we actually need: Analyzing object detection performance using synthetic and real data," 2019, *arXiv:1907.07061*.

[3] T. Linder, K. Y. Pfeiffer, N. Vaskevicius, R. Schirmer, and K. O. Arras, "Accurate detection and 3D localization of humans using a novel YOLO-based RGB-D fusion approach and synthetic training data," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 1000–1006.

[4] J. Tremblay *et al.*, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 969–977.

[5] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 23–30.

[6] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; Increasing shape bias improves accuracy and robustness," in *Proc. Int. Conf. Learn. Representations*, 2019. [Online]. Available: openreview.net

[7] E. Santana and G. Hotz, "Learning a driving simulator," *CoRR*, 2016, *arXiv:1608.01230*.

[8] S. I. Nikolenko, *Synthetic Data for Deep Learning*. Springer International Publishing, 2021.

[9] B. Taha and A. Shoufan, "Machine learning-based drone detection and classification: State-of-the-art in research," *IEEE Access*, vol. 7, pp. 138669–138682, 2019.

[10] M. Vrba and M. Saska, "Marker-less micro aerial vehicle detection and localization using convolutional neural networks," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2459–2466, Apr. 2020.

[11] A. Barisic, M. Car, and S. Bogdan, "Vision-based system for a real-time detection and following of UAV," in *Proc. Workshop Res., Educ. Develop. Unmanned Aerial Syst.*, 2019, pp. 156–159.

[12] A. Yavariabdi, H. Kusetogullari, T. Celik, and H. Cicek, "FastUAV-NET: A multi-UAV detection algorithm for embedded platforms," *Electronics*, vol. 10, Mar. 2021, Art. no. 724.

[13] P. M. Wyder *et al.*, "Autonomous drone hunter operating by deep learning and all-onboard computations in GPS-denied environments," *PLoS One*, vol. 14, Nov. 2019, Art. no. e0225092.

[14] Y. Chen, P. Aggarwal, J. Choi, and C.-C. J. Kuo, "A deep learning approach to drone monitoring," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2017, pp. 686–691.

[15] C. Aker and S. Kalkan, "Using deep networks for drone detection," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, 2017, pp. 1–6.

[16] M. Denninger *et al.*, "BlenderProc: Reducing the reality gap with photo-realistic rendering," in *Proc. Robot.: Sci. Syst. (RSS) Workshops*, 2020.

[17] J. Peng, C. Zheng, T. Cui, Y. Cheng, and L. Si, "Using images rendered by PBRT to train faster R-CNN for UAV detection," in *Proc. Int. Conf. Central Europe Comput. Graph., Visualization Comput. Vis.*, Západočeská Univerzita, 2018, pp. 13–18,.

[18] L. Liu *et al.*, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, pp. 261–318, Oct. 2019.

[19] A. Barisic, F. Petric, and S. Bogdan, "Brain over Brawn - Using a stereo camera to detect, track and intercept a faster UAV by reconstructing its trajectory," 2021, *arXiv:2107.00962*.

[20] Y. Adini, Y. Moses, and S. Ullman, "Face recognition: The problem of compensating for changes in illumination direction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 721–732, Jul. 1997.

[21] "CGTrader - 3D models for VR / AR and CG projects," 2021. Accessed: Sep. 9, 2021. [Online]. Available: https://www.cgtrader.com/

[22] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "RotorS—A modular gazebo MAV simulator framework," in *Studies in Computational Intelligence*. Berlin, Germany: Springer, 2016, pp. 595–625.

[23] "PolyHaven - The public 3D asset library," 2021. Accessed: Sep. 9, 2021. [Online]. Available: https://polyhaven.com/

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[25] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[27] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," 2019. [Online]. Available: https://github.com/facebookresearch/detectron2

[28] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2014, pp. 740–755.

[29] K. Kwok, G. Webster, A. Athalye, and L. Engstrom, "TensorFire - Blazing-fast in-browser neural networks," 2021. Accessed: Nov. 20, 2021. [Online]. Available: https://tenso.rs/demos/fast-neural-style/