Data augmentation for vehicle detection with diffusion-based object inpainting

Sebastiaan P. Snel[a,b], Thijs A. Eker[a], Ella P. Fokkinga[a], Arnoud Visser[b], Klamer Schutte[a], and Friso G. Heslinga[a]

[a]TNO - Intelligent Imaging, Oude Waalsdorperweg 63, the Hague, the Netherlands
[b]University of Amsterdam, Science Park 900, Amsterdam, the Netherlands

ABSTRACT

Automated vehicle detection in video footage captured by Unmanned Aerial Vehicles (UAVs) is a critical capability in security and defense domains, especially for environments where communication is jammed. Development of deep learning-based object detectors for this purpose typically requires large-scale datasets, which can be hard to obtain due to limited access to relevant environments. To address this challenge, synthetic data has been proposed as a supplementary source of training data, introducing additional variations in the appearance and positioning of objects. One promising strategy for generating synthetic data is inpainting, where objects of interest are seamlessly integrated into various backgrounds. However, traditional inpainting techniques lack spatial and contextual awareness, limiting their effectiveness for data augmentation. Recent advancements in generative AI, specifically diffusion models, have demonstrated improvements in object harmonization and spatial control for object inpainting, enabling realistic foreground-background matching with a high level of diversity. In this work, we explore the value of diffusion-based inpainting as a data augmentation technique. We use the inpainting model AnyDoor to enrich a small subset (1000 frames), of the VisDrone train dataset with inpainted versions of minority-class objects (buses, vans, trucks). We train YOLOX detectors on datasets with increasing amounts of synthetic vehicles (1x, 5x, 10x, and 20x) and analyze the impact on detection performance. Results show that zero-shot inpainting can substantially improve detection for buses up to an augmentation factor of 10x, with no improvements at 20x. Effects for vans and trucks are mixed and sometimes negative. Fine-tuning AnyDoor provided limited additional benefit under the tested conditions. Overall, diffusion-based inpainting shows potential as a data augmentation strategy in low-resource UAV scenarios. Future work should explore strategies to increase contextual diversity, such as adding multiple synthetic objects per image or incorporating automated quality control for synthetic samples.

Keywords: Inpainting; Diffusion; Data augmentation; Generative AI; Synthetic data; Deep learning; Object detection; Vehicle detection

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are increasingly deployed in security and defense for intelligence, surveillance, and reconnaissance operations. A key capability in these missions is the automated detection and classification of vehicles in aerial video footage.[1–3] Automating this step reduces the operator's workload and is critical in scenarios where video transmission is interrupted due to jamming.

Training robust deep learning-based detectors requires large datasets, that are diverse and representative of real-world conditions. Collecting such datasets is challenging due to restricted access to operational environments and the limited availability of certain object classes, particularly in rare configurations, such as unusual poses, occlusions, or lighting conditions. This lack of diversity makes object detection models prone to overfitting on training data and reduces their ability to generalize to new scenarios.[4] To address these limitations, synthetic data has been proposed as a supplementary source for training models, introducing additional variations in the object appearance and positioning. Previous methods often rely on 3D-model simulation pipelines,[5–9] but these methods are time-consuming to develop and suffer from domain gaps when applied on real-world data.[10,11] An

Corresponding author: Friso G. Heslinga. E-mail: fgheslinga@gmail.com

Artificial Intelligence for Security and Defence Applications III, edited by Hugo J. Kuijf, Radhakrishna Prabhu, Yitzhak Yitzhaky, Proc. of SPIE Vol. 13679, 136790V
2025 Published by SPIE · 0277-786X · doi: 10.1117/12.3070068

alternative strategy is inpainting, where objects of interest are inserted into various backgrounds to create new samples.[12–14] However, traditional inpainting techniques lack semantic understanding, struggle with large or complex missing regions, and often produce visible artifacts. These limitations reduce their effectiveness as a data augmentation strategy.

Recent advances in generative AI provide new opportunities to synthesize image data to train AI models for automated scene understanding.[15] Specifically, diffusion-based inpainting models have demonstrated improvements in three key areas: blending objects naturally into their surroundings (object harmonization), preserving the identity of the reference object rather than generating a similar alternative (ID consistency), and increasing appearance diversity - all while retaining a high level of spatial control.[16–18] Recent studies highlight the effectiveness of reference image-based inpainting for developing medical segmentation models.[19,20] Previous work has explored inpainting for air traffic detection, based on text-prompts.[21] Despite these studies, the potential of diffusion-based inpainting for UAV-based vehicle detection remains largely unexplored. Addressing this gap is critical for improving detection of rare vehicle classes without costly or infeasible real-world data collection.

In this paper, we study whether diffusion-based inpainting can serve as an effective data augmentation technique for vehicle detection from a UAV perspective. Specifically, we use the diffusion-based inpainting model AnyDoor[16] to extend a small subset of the VisDrone training dataset[22] with inpainted versions of minority-class vehicles (underrepresented vehicle categories). We compare zero-shot and fine-tuned inpainting performance, followed by training YOLOX object detection models[23] on datasets augmented with varying quantities of the inpainted images. Section 2 provides an overview of generative AI inpainting techniques and the use of synthetic data as data augmentation. Section 3 describes our methodology; Section 4 presents results for both the inpainting and object detection; and Section 5 concludes with the key findings.

2. RELATED WORKS

2.1 Diffusion models

Beyond traditional augmentation techniques such as geometric transformations and color adjustments, current research increasingly explores generative AI-based image synthesis to enrich training datasets.[24–27] Diffusion models[28–30] have made progress in becoming state-of-the-art for image synthesis, demonstrating superior data diversity and fidelity over prior models such as GANs[31,32] and VAEs.[33] At the heart of these models lies a diffusion mechanism that incrementally adds noise to the input data until the resulting samples are indistinguishable from pure noise. During inference, a denoiser trained to reverse the noise addition process is used to progressively refine pure noise samples until they resemble clean data. Beyond image synthesis, diffusion models have also been employed for the generation of video,[34] motion,[35,36] and audio.[37,38] Widely used diffusion frameworks, such as Stable Diffusion,[30] DALL-E2,[39] and FLUX[40] are primarily designed for text-to-image generation. These models use a text encoder like CLIP[41] to transform a text prompt into a latent embedding, which conditions the denoising process.

While text-to-image models generate visually realistic images, they often lack control over spatial composition, object layout, and pose. In 2023, ControlNet[42] introduced additional conditioning signals, such as edge maps, depth maps, segmentations, and poses.[43,44] ControlNet enhances a pretrained diffusion model by duplicating its weights into two branches: a frozen backbone that preserves learned representations and a trainable copy, enabling precise control over the generation process. In low-resource vision domains, pretrained diffusion models frequently lack detailed knowledge of specific classes, leading to unrealistic image generation or low-diversity outputs. This can be addressed by providing a reference image to guide the process, for example using a ControlNet building block.[16,45]

2.2 Inpainting

Image inpainting techniques replace or insert visual elements into background images to achieve seamless blends. Early approaches used GANs to treat inpainting as a conditional generation task,[46,47] integrating high-level recognition and pixel synthesis via adversarial loss encoder-decoder networks.[48] While effective, these methods often suffer from boundary artifacts and identity inconsistency.

To improve control, example-guided methods like Paint by Example (PbE)[18] merge reference and source images using binary masks. PbE compresses the reference image into a 1024-dimensional vector via a CLIP encoder, preserving semantics while avoiding direct copy-paste. However, this bottleneck limits low-level detail synthesis, causing identity distortion.[17] PhD (Paste, Inpaint and Harmonize via Denoising)[17] addresses this by extracting subjects with SAM,[49] pasting them onto backgrounds, and harmonizing via a self-supervised inpainting model. Unlike PbE, PhD avoids fine-tuning large diffusion models and retains strong composition abilities. For fine-grained identity preservation, "An image is worth one word"[50] uses textual inversion to learn pseudo-words from three to five user-provided images, enabling synthesis of objects in new styles while retaining the traits that identify the individual. DreamBooth[51] builds on this by fine-tuning Stable Diffusion[30] with a class-specific prior loss, allowing diverse scene synthesis.

AnyDoor[16] enables zero-shot, subject-driven inpainting by learning object transformations over time from video datasets. During training, one frame provides the target object while another serves as supervision. The method employs adaptive time-step sampling to optimize the denoising process: early steps focus on structural generation, while later steps refine textures and colors. Although trained on video data, AnyDoor can be applied to both video and image inpainting. For video, the model allocates 50% more sampling to early steps, to maintain structural consistency across frames. For single images, later steps are emphasized to enhance fine-grained detail synthesis. Users can guide subject placement by drawing a mask on the background image. AnyDoor integrates a ControlNet architecture[42] with the DINOv2 feature extractor,[52] enabling high-quality synthesis. This combination of temporal and spatial control makes AnyDoor a robust foundation for our research pipeline.

2.3 Image synthesis for data augmentation

Previous work illustrates that language-based diffusion models can effectively augment labeled classification datasets.[26,53–55] When expanding datasets with synthetic images, maintaining a balance between real and synthetic data is crucial, as synthetic data may exaggerate certain features or inherit biases from the generative model.[54] One approach to address this is assigning sampling probabilities to real and synthetic images to mitigate imbalance.[53] To reduce bias, synthetic data should include sufficient variation.[5] For instance, Odgen[55] enhances variety in synthetic classification datasets by using object-wise conditioning modules that control object categories and placement. Although this method fine-tunes pretrained diffusion models on domain-specific datasets, the study acknowledges that generating high-quality synthetic images for novel domains remains challenging.

X-paste[56] employs two strategies to obtain additional image instances of a target object with diverse appearances, viewpoints, and styles: (1) generating synthetic data using the zero-shot SD1.4 text-to-image diffusion model,[30] and (2) scraping real images from the internet. To filter the scraped images for quality, the semantic similarity between each category and the images is evaluated using a CLIP model.[41] The baseline for evaluating these strategies is object detection using CenterNet2[57] on the LVIS dataset.[58] Synthetic instances achieve a box AP of 36.3 and a mask AP of 32.3, outperforming the baseline by +1.9 and +1.5 mAP, respectively. Improvements are especially notable for rare categories, with gains of +4.0 box AP and +3.6 mask AP. Scaling synthetic data from 100,000 to 300,000 further enhances performance. For comparison, 300,000 real images scraped from the internet yield a +2.3 box AP and +1.9 mask AP. These results demonstrate the potential of generative data augmentation and underscore its relevance to our research.

Diversity in datasets is essential for robust object detection,[5] as models must generalize across variations in lighting, weather, and environmental conditions. In the vehicle-focused research, such as the automotive or military applications, factors such as vehicle speed, road orientation, and vehicle lighting are particularly relevant. Petersen et al.[59] address this challenge by introducing scene-aware object synthesis for data augmentation. They propose a probabilistic location model that predicts realistic object placement within existing scenes. Once the vehicle location is determined, object synthesis is performed by diffusion-based inpainting using an SD2 model[30] with ControlNet.[42] This scene-aware approach improves Faster R-CNN[60] object detection performance by +1.6 mAP on the BDD100K dataset.[61] These findings support our motivation to explore object detection enhancements through the AnyDoor[16] generative inpainting pipeline, specifically using vehicle data from the VisDrone-19 drone perspective dataset.[22]

3. METHODS

Parts of this manuscript were drafted with the assistance of Microsoft Copilot with GPT-5, to improve clarity and phrasing. All content was reviewed and verified by the authors, who take full responsibility for the final manuscript.

3.1 Dataset

The experiments were conducted with the VisDrone-19 dataset,[22] a large-scale benchmark for object detection and tracking in drone imagery. We focused exclusively on four vehicle classes: cars, buses, trucks, and vans, while we ignored all other object categories during training and evaluation. For training AnyDoor, we selected 50 videos from the VisDrone-19-VID train dataset, which together contain 402 unique cars, 44 buses, 99 trucks, and 152 vans. To train an object detector, we randomly sampled 1,000 frames from these 50 videos, ensuring that each frame contained at least one instance of the target classes, creating the Visdrone1000 dataset. Evaluation and validation were performed on the official VisDrone-19-DET validation and test sets without further modification.

3.2 Fine-tuning AnyDoor

We used AnyDoor[16] to generate additional training data for object detection. In the VisDrone-19 dataset, the vehicle classes bus, van, and truck are underrepresented in comparison to the car class. With AnyDoor, we take the bounding box of a car and replace it with, for example, a synthetic van sourced from a different video frame. This operation can be performed zero-shot, since AnyDoor already demonstrates reasonable out-of-the-box performance for vehicle inpainting tasks. Figure 1 illustrates how a synthetic image can be created using this approach.

Because AnyDoor was originally trained on generic image datasets and not on drone-view imagery, we hypothesized that fine-tuning it on VisDrone videos would yield more realistic and context-appropriate inpainting results. Fine-tuning requires access to video sequences because AnyDoor's training principle relies on using an earlier frame as the reference, and inpainting the object into a later frame from the same sequence. Specifically, we trained three separate AnyDoor models: one for buses, one for vans, and one for trucks, to maximize the fidelity of the generated samples for each class.

Each AnyDoor model is trained for 30 epochs, where one epoch consists of $5,000$ inpainting steps. In each step, a frame was randomly selected as the reference, and the object it contains was inpainted into another frame from the same video sequence: the target. We selected a batch size of 8 with gradient accumulation over 4 steps (with an effective batch size of 32), a learning rate of 1×10^{-5}, and mixed precision (FP16) on an NVIDIA A40 GPU. The training objective follows a linear noise schedule from 0.00085 to 0.0120 over $1,000$ diffusion time steps. For more details on all other hyperparameters, see the original AnyDoor paper.[16]

3.3 Synthetic data generation

We tested two approaches to generate synthetic data: one using the zero-shot (*zs*) AnyDoor model and one using the corresponding fine-tuned (*ft*) model.

For each target class, we created augmented datasets by generating synthetic instances at four augmentation factors: 1×, 5×, 10×, and 20× the original number of annotated objects in VisDrone1000. Across the 1,000 frames in the dataset, the original annotation counts were: *buses* - 417 bounding boxes (44 unique buses), *trucks* – 1,036 bounding boxes (99 unique trucks), and *vans* – 2,192 bounding boxes (152 unique vans). In the target frame, we replace one of the five largest bounding boxes with a synthetic vehicle sourced from a reference frame (see Figure 1). This choice ensures sufficient spatial resolution for the inpainted object. For higher augmentation factors (5×, 10×, 20×), or for classes with more than 1,000 original annotations (trucks and vans), the same set of 1,000 target images was reused multiple times, each time inserting a different synthetic instance. As an example, for the bus class, the *bus*1× dataset was created by generating 417 synthetic buses - matching the original number of bus annotations - and inpainting them into 417 randomly selected target images (as illustrated in Figure 1). In addition to class-specific datasets, we created combined datasets that include synthetic instances of all three classes (buses, trucks, vans) at the same augmentation factor. For example, the *combined*10× dataset contains 10× the original number of buses, 10× trucks, and 10× vans.

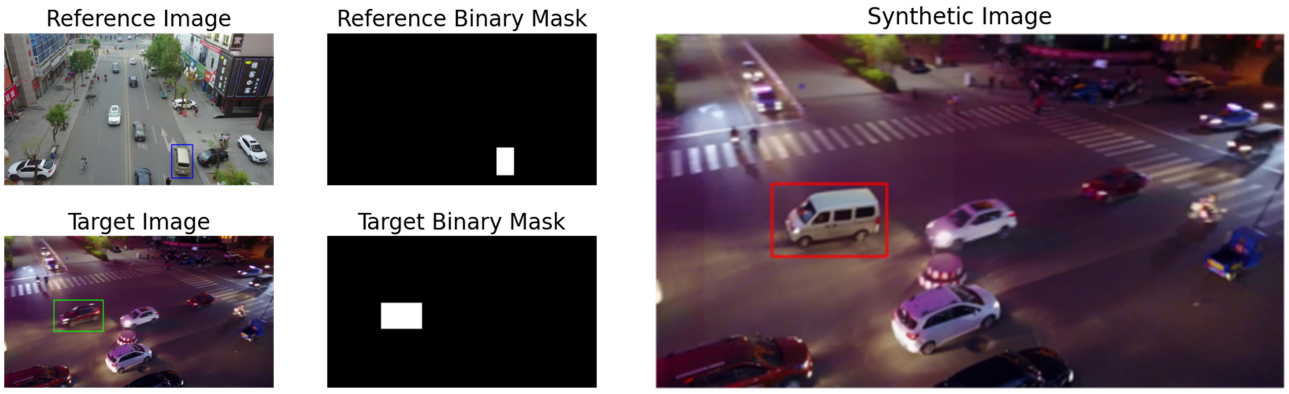

Figure 1: Reference, target, and binary mask images, with the synthetic vehicle shown in the red bounding box on the right. A van is selected in the reference image, denoted by the blue bounding box, which results in the reference binary mask. In the target image, a car is selected, indicated by the green bounding box, which results in the target binary mask. The van from the reference image is inpainted at the target binary mask location using AnyDoor,[16] resulting in the synthetic image on the right.

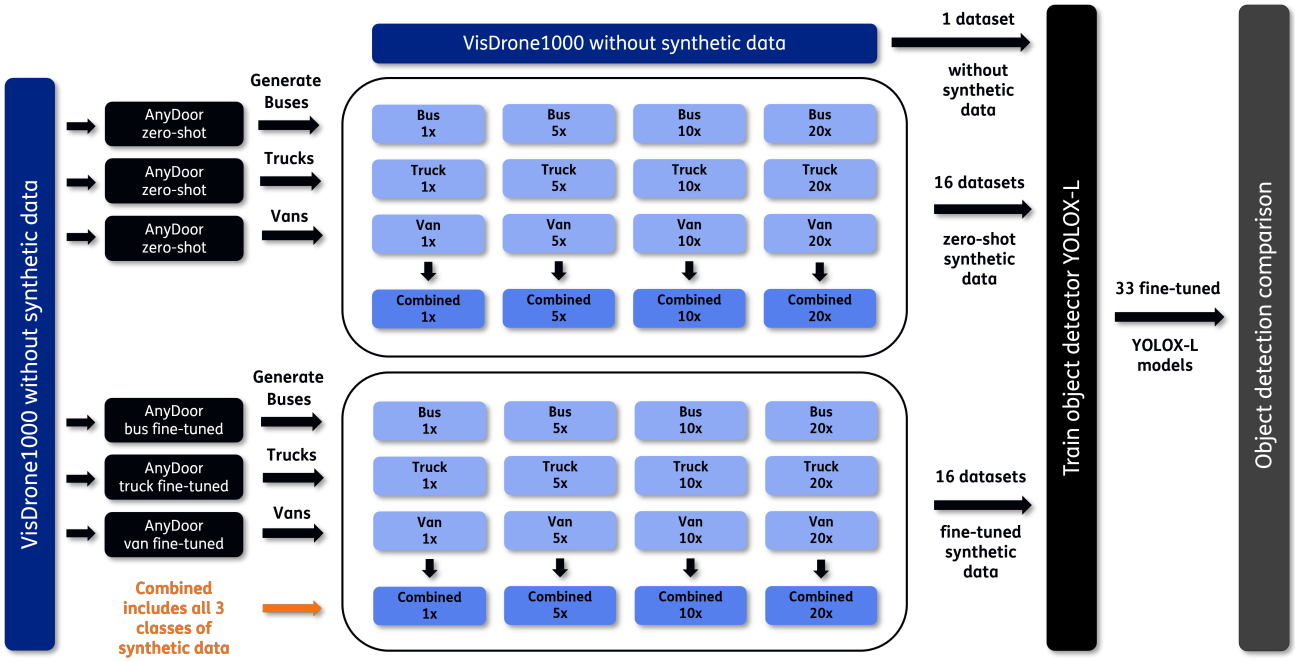

Figure 2: Overview of the experimental setup for synthetic data generation and model training. Synthetic buses, trucks, and vans were generated using AnyDoor in two variants: zero-shot (top) and fine-tuned (bottom). For each class, vehicle class-specific datasets were created at four augmentation factors (1×, 5×, 10×, 20×) relative to the original number of annotated objects. The combined datasets include synthetic buses, trucks, and vans at the same augmentation factor (e.g. combined 10× includes 10× buses, 10× trucks, and 10× vans). These datasets were used to train YOLOX object detectors, whose performance was compared against a baseline trained on VisDrone1000 without synthetic data.

In total, this approach yields 32 datasets: 16 datasets generated with zero-shot AnyDoor, and 16 datasets generated with the fine-tuned models. See Figure 2 for an overview of the generated datasets.

3.4 YOLOX object detection

To evaluate the effectiveness of the AnyDoor-generated datasets, we used the 32 synthetic datasets and the original VisDrone1000 dataset to fine-tune YOLOX-L (referred to as YOLOX in the remainder of this paper)[23]

which is pretrained on the COCO dataset.[62] As a baseline, YOLOX was trained solely on the VisDrone1000 dataset, which contains only real images. For each of the 32 synthetic datasets (Figure 2), we trained YOLOX on the original VisDrone1000 data augmented with the corresponding synthetic images. We evaluated YOLOX on the VisDrone-DET test-dev split, reporting mAP@25, size-wise AP@25 for small, medium, and large objects, and per-class AP for car, van, truck, and bus. Each training configuration was repeated three times, and we report the mean and standard deviation of AP values.

We trained YOLOX for approximately 33,000 iterations in each experiment, using Stochastic Gradient Descent (SGD) with a base learning rate of 0.02, momentum of 0.9, and weight decay of 5×10^{-4}. For the baseline VisDrone1000 dataset, this corresponds to around 100 epochs. When training on augmented datasets, containing more samples, the number of epochs is proportionally reduced to maintain a similar total number of training iterations. This approach ensures that performance differences are attributable to the composition of the training data rather than differences in training time, and it allows us to keep computational costs consistent across experiments. Although the number of epochs is reduced, real VisDrone images are, in fact, not seen fewer times using this approach: each synthetic frame is based on an original VisDrone frame with an additional inpainted object. We employed a quadratic warm-up schedule for the first five epochs, cosine annealing until epoch 90, and a constant learning rate for the final 10 epochs. To accommodate the memory constraints of the NVIDIA A40 GPU (48 GB), we set the batch size to 3 and train at a high-resolution input size of 1920×1920, enhancing detection performance for small objects common in drone imagery. Advanced data augmentations such as Mosaic,[63] mixup,[64] RandomAffine, HSV augmentation, and random horizontal flipping were applied to improve generalization; however, these augmentations were disabled in the final 10 epochs to allow the model to adapt to the real data distribution.

4. RESULTS

This section describes the synthetic data generated by AnyDoor, followed by an evaluation of YOLOX performance on the VisDrone-DET test set.

4.1 Synthetic data generation

Figure 3 illustrates representative synthetic outputs from zero-shot AnyDoor and fine-tuned models at different epochs. While the differences are slight, there appears to be reduction of artifacts and improved object background harmonization over epochs.

4.2 YOLOX object detection

Table 1 reports the mAP@25 scores of the YOLOX object detector trained on the different datasets, evaluated across three bounding box sizes and four vehicle classes.

Baseline. Figure 4 illustrates representative detections on real VisDrone test images using the YOLOX model trained on the original VisDrone1000 dataset. The model predominantly detects cars, reflecting the class imbalance in the training set, while predictions for underrepresented classes such as buses, trucks, and vans are less frequent and occasionally include false positives.

Training on the real-only VisDrone1000 subset yields an overall mAP@25 of 23.7, with size-wise AP of 19.1 (small), 29.5 (medium), and 21.4 (large). Per-class AP is 60.0 (car), 17.2 (van), 4.9 (truck), and 12.5 (bus). These values serve as the reference for all comparisons below.

Effect of synthetic augmentation. Table 1 summarizes the relative changes with respect to the baseline for all 33 configurations. Several patterns can be seen:

(i) *Buses:* Synthetic augmentation strongly benefits bus detection overall, but fine-tuning does not significantly outperform zero-shot. Zero-shot at 10× yields the largest overall gain (+2.5) and a +4.3 increase for buses, with improvements concentrated on medium and large objects. Fine-tuned variants achieve similar results; even at 20×, bus AP rises by +2.3, but without a clear advantage over zero-shot.

Figure 3: Examples of synthetic buses, trucks, and vans generated by zero-shot AnyDoor and fine-tuned models at different epochs.

Figure 4: YOLOX predictions on real VisDrone test images. Blue bounding boxes indicate cars, orange indicate trucks, bright yellow indicate vans, and dark yellow indicate buses. Red dashed boxes highlight examples of false positive predictions.

(ii) *Trucks:* Zero-shot augmentation has a limited effect and becomes slightly negative at 20×. Fine-tuned models improve truck AP at moderate scales (1–5×, up to +5.9) and provide modest overall AP gains. Larger synthetic volumes ($10 - 20\times$) reduce overall performance compared to smaller scales.

(iii) *Vans:* Augmenting vans generally reduces performance in both zero-shot and fine-tuned settings. Zero-shot augmentation consistently lowers overall AP, while fine-tuned models perform somewhat better but still do not achieve improvements over the baseline.

Error patterns. To gain more insights into these trends, Figure 5 presents the baseline confusion matrix and those from fine-tuned configurations where 5× synthetic buses, trucks, or vans were added. Across all augmented runs, the synthetic class shows an increase in correct predictions, confirming that augmentation improves recall for the targeted class. However, side effects differ by class. For buses, the main change is a reduction in background predictions, while misclassification of cars as buses remains at zero in both baseline and augmented runs. For vans, the confusion with cars increases notably: from 2% in the baseline to 6% after adding synthetic vans. Trucks show a similar pattern, with car-to-truck confusion rising from 0% to 4%, and more background being classified as trucks.

5. DISCUSSION

The objective of this study was to evaluate whether generative inpainting with AnyDoor can serve as an effective data augmentation strategy in a low-resource scenario for drone-based object detection. To simulate such conditions, we restricted training to a subset of the VisDrone dataset (VisDrone1000), containing only 1,000 images. The baseline YOLOX model trained on this limited set achieves an overall mAP@25 of 23.7, with strong performance for cars (AP = 60.0) but substantially lower AP for less frequent classes such as vans (17.2), trucks (4.9), and buses (12.5). This highlights how difficult it is to train accurate detectors when only a small amount of data is available.

5.1 Fine-tuning versus zero-shot AnyDoor

Fine-tuning AnyDoor did not consistently outperform zero-shot usage in our experiments, for any of the classes excepts vans. Downstream detection results showed no clear advantage for fine-tuned models, and additional experiments using image-quality metrics (FID, Inception Score, and DINO similarity) revealed no measurable differences after fine-tuning. These metrics were therefore not reported in this paper. It should be noted that such metrics may not fully capture inpainting quality in this context, and that developing suitable evaluation measures was beyond the scope of this study. Overall, these findings suggest that fine-tuning, under the conditions tested, provides limited benefit. A likely explanation is that the base AnyDoor model already possesses knowledge of vehicle classes such as buses, vans, and trucks, which are represented in its pretraining data. This situation is different for military applications and, therefore, fine-tuning is expected to play a more critical role.

5.2 Impact of synthetic augmentation by class and object size

Augmenting buses consistently improved detection performance, with zero-shot augmentation at 10× giving the largest overall gain (+2.5) and a +4.3 increase for buses. In contrast, vans showed no improvement in overall mAP under any configuration, and van-specific AP only improves for the finetuned setting. Visual inspection of synthetic samples did not reveal systematically worse inpainting for vans. It is more likely that the van's strong visual similarity to cars, which is the most prevalent class in the test set, induced confusion, as supported by the confusion matrices (Figure 5), which show increased misclassification of cars and background as vans in van-augmented runs.

Performance gains were most pronounced for large objects (Table 1), which aligns with our augmentation strategy: synthetic objects were always inserted into one of the five largest bounding boxes in each background image. This choice was made to preserve sufficient spatial resolution for the inpainted object, as smaller bounding boxes would result in loss of detail. However, this can introduce a bias toward larger objects, which explains why performance gains are more pronounced for large bounding boxes (Table 1). Future work could explore strategies that include smaller bounding boxes to achieve a more balanced effect across object sizes.

Subset	mAP@25	Bbox-s	Bbox-m	Bbox-l	Cars	Vans	Trucks	Buses
baseline	23.7 ± 0.1	19.1 ± 0.0	29.5 ± 0.3	21.4 ± 1.4	60.0 ± 0.2	17.2 ± 0.2	4.9 ± 0.5	12.5 ± 0.7
bus1×-zs	+1.6 ± 0.2	+0.9 ± 0.2	+1.8 ± 0.2	+0.9 ± 0.6	+0.9 ± 0.3	+0.4 ± 0.6	+0.3 ± 0.3	+5.2 ± 0.0
bus5×-zs	+1.9 ± 0.6	+0.6 ± 0.3	+2.1 ± 0.6	+2.0 ± 1.7	+1.5 ± 0.1	+1.5 ± 0.9	+0.4 ± 0.7	+4.2 ± 0.9
bus10×-zs	+2.5 ± 0.7	+0.9 ± 0.4	+2.5 ± 1.0	+4.9 ± 0.3	+2.0 ± 0.5	+2.8 ± 0.4	+1.1 ± 0.5	+4.3 ± 1.3
bus20×-zs	+0.7 ± 2.5	+0.1 ± 0.7	-0.3 ± 3.4	+2.0 ± 5.5	+1.4 ± 2.0	+1.1 ± 0.9	+0.6 ± 1.3	-0.1 ± 5.7
truck1×-zs	-0.1 ± 0.3	+0.1 ± 0.6	0.0 ± 0.1	+0.9 ± 0.5	-0.9 ± 0.7	+0.5 ± 0.7	+1.4 ± 0.7	-1.4 ± 0.4
truck5×-zs	-0.4 ± 1.3	-0.7 ± 1.7	-1.0 ± 1.4	+0.4 ± 1.1	-0.7 ± 2.5	-0.2 ± 1.4	+0.5 ± 1.3	-1.0 ± 0.8
truck10×-zs	+0.2 ± 0.4	+0.1 ± 0.1	-0.4 ± 0.5	+0.6 ± 1.0	+1.3 ± 0.3	+0.6 ± 0.5	+1.3 ± 0.2	-2.4 ± 1.5
truck20×-zs	-2.7 ± 0.4	-2.9 ± 0.2	-3.3 ± 0.8	-2.0 ± 1.9	-5.1 ± 0.7	-2.5 ± 0.8	+0.3 ± 0.5	-3.5 ± 0.8
van1×-zs	-3.4 ± 0.6	-1.6 ± 2.0	-4.8 ± 0.6	-2.3 ± 0.4	-4.0 ± 1.9	-2.1 ± 1.1	-1.9 ± 0.7	-5.3 ± 1.6
van5×-zs	-2.0 ± 0.5	-1.0 ± 0.2	-3.0 ± 0.7	-1.3 ± 0.5	-0.5 ± 0.4	+0.3 ± 0.9	-1.1 ± 0.0	-6.3 ± 1.0
van10×-zs	-5.9 ± 0.5	-4.9 ± 0.4	-6.8 ± 0.7	-3.8 ± 0.7	-9.7 ± 1.2	-3.9 ± 0.8	-2.7 ± 0.5	-7.0 ± 1.4
van20×-zs	-8.3 ± 0.3	-6.9 ± 0.0	-10.0 ± 0.5	-7.4 ± 0.6	-14.5 ± 0.6	-6.0 ± 0.3	-3.3 ± 0.4	-9.1 ± 1.3
combined1×-zs	-0.5 ± 0.9	+0.6 ± 0.2	-1.3 ± 1.3	+1.2 ± 2.7	-1.5 ± 1.2	+1.0 ± 0.1	+1.3 ± 0.6	-2.5 ± 2.1
combined5×-zs	-1.1 ± 0.0	-1.9 ± 0.3	-0.9 ± 0.2	+1.1 ± 0.7	-5.7 ± 0.8	-0.9 ± 0.3	+1.4 ± 0.2	+0.8 ± 0.8
combined10×-zs	-5.3 ± 0.2	-4.9 ± 0.3	-6.3 ± 0.5	-4.6 ± 1.3	-12.3 ± 0.3	-2.9 ± 0.8	-1.1 ± 0.3	-4.6 ± 0.5
combined20×-zs	-5.3 ± 0.3	-4.6 ± 0.6	-6.4 ± 0.2	-6.8 ± 0.5	-11.1 ± 0.9	-3.9 ± 0.3	-2.1 ± 0.3	-3.8 ± 0.2
bus1×-ft	+0.2 ± 2.1	-0.3 ± 2.1	-0.3 ± 2.7	+1.8 ± 2.3	-1.0 ± 2.3	-0.9 ± 2.0	-0.2 ± 1.4	+3.2 ± 3.0
bus5×-ft	+2.1 ± 0.8	+1.4 ± 0.2	+2.2 ± 1.1	+1.1 ± 0.9	+1.9 ± 0.6	+1.6 ± 0.8	+1.2 ± 0.3	+3.9 ± 2.7
bus10×-ft	+2.2 ± 0.2	+1.5 ± 0.2	+1.8 ± 0.3	+4.1 ± 0.6	+2.3 ± 0.2	+1.7 ± 0.2	+0.8 ± 0.3	+4.3 ± 0.9
bus20×-ft	+1.8 ± 1.1	+1.3 ± 0.4	+1.4 ± 1.3	+5.0 ± 2.4	+2.4 ± 1.2	+1.5 ± 1.9	+1.3 ± 0.4	+2.3 ± 1.6
truck1×-ft	+1.3 ± 0.7	+1.0 ± 1.2	+1.2 ± 1.0	+1.9 ± 1.2	0.0 ± 0.8	+0.1 ± 1.4	+5.2 ± 1.3	0.0 ± 1.4
truck5×-ft	+2.8 ± 0.8	+1.9 ± 0.4	+2.6 ± 1.1	+4.8 ± 0.4	+2.3 ± 0.2	+1.1 ± 1.1	+5.9 ± 0.4	+2.0 ± 2.0
truck10×-ft	-2.6 ± 4.0	-1.1 ± 2.0	-4.1 ± 5.7	-6.4 ± 7.8	-4.6 ± 6.3	-1.7 ± 3.4	+2.2 ± 3.4	-6.0 ± 3.1
truck20×-ft	-2.2 ± 0.4	-3.3 ± 0.6	-1.9 ± 0.3	+1.0 ± 0.4	-5.7 ± 0.8	-1.6 ± 0.5	+2.2 ± 0.6	-3.3 ± 0.1
van1×-ft	-0.3 ± 0.2	+0.8 ± 0.2	-0.7 ± 0.3	+0.9 ± 0.8	+0.5 ± 0.2	+3.1 ± 0.4	-0.7 ± 0.2	-4.1 ± 0.3
van5×-ft	-0.5 ± 0.2	-0.1 ± 0.3	-1.0 ± 0.2	-1.6 ± 0.6	-0.6 ± 0.4	+4.5 ± 0.5	-0.2 ± 0.1	-5.5 ± 0.5
van10×-ft	-3.7 ± 0.5	-2.5 ± 0.9	-4.5 ± 0.1	-3.5 ± 1.5	-7.6 ± 0.9	+1.0 ± 0.9	-1.3 ± 0.4	-6.6 ± 1.3
van20×-ft	-7.3 ± 0.2	-5.5 ± 0.2	-9.0 ± 0.5	-7.7 ± 0.8	-14.3 ± 0.8	-2.9 ± 0.4	-2.6 ± 0.3	-9.1 ± 0.2
combined1×-ft	+3.4 ± 0.3	+3.3 ± 0.5	+3.4 ± 0.4	+3.0 ± 1.0	+2.0 ± 0.3	+5.1 ± 1.0	+6.0 ± 1.1	+0.5 ± 0.4
combined5×-ft	+1.2 ± 0.4	+0.4 ± 0.3	+1.9 ± 0.6	+2.2 ± 0.8	-3.5 ± 0.8	+4.4 ± 0.3	+4.5 ± 0.4	-0.4 ± 1.5
combined10×-ft	-5.2 ± 0.3	-5.0 ± 0.2	-5.9 ± 0.3	-4.1 ± 0.4	-13.3 ± 0.6	-1.3 ± 0.0	+0.3 ± 0.3	-6.1 ± 1.0
combined20×-ft	-0.9 ± 0.2	-1.1 ± 0.1	-1.0 ± 0.4	-2.1 ± 1.4	-6.0 ± 0.6	+3.4 ± 0.4	+1.7 ± 0.2	-2.6 ± 0.7

Table 1: Overview of the mAP@25 scores in percentages (with standard deviations) for the baseline YOLOX detector and the detector trained on the synthetic datasets. The metrics are reported for three bounding box sizes: small (bbox-s), medium (bbox-m), and large (bbox-l), and four vehicle classes. "*zs*" and "*ft*" respectively denote zero-shot and fine-tuned AnyDoor inpainting. The first column lists the dataset used for training, where the number following the vehicles class indicates the augmentation factor: e.g. for "bus1×-zs" means that number of synthetic zero-shot in-painted buses equals the number of real buses in the dataset, "bus5×-zs" means five times as mine, and so on. All reported values for the synthetic datasets represent the change in mAP@25 compared to the baseline model.

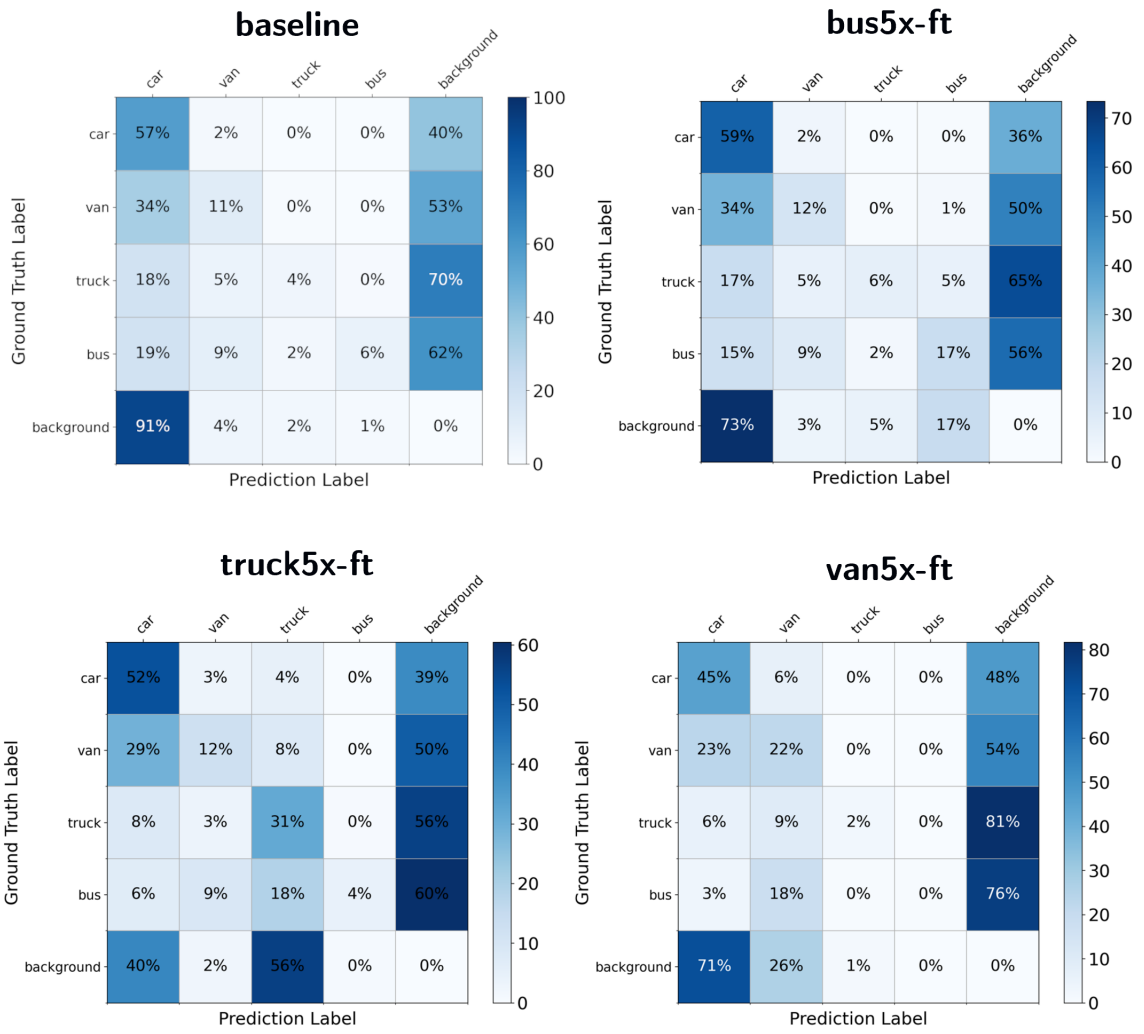

Figure 5: Confusion matrices for the baseline YOLOX model and three fine-tuned models trained with synthetic buses, trucks, or vans at augmentation factor 5×. Values represent percentages of predictions for each class (car, van, truck, bus, background). Color scales differ between matrices to highlight relative differences in performance rather than absolute values.

5.3 Effect of augmentation factor

Increasing synthetic data beyond a certain point does not guarantee continuous performance gains. For buses, performance saturates after 10×, while for trucks and vans, large-scale augmentation ($10 - 20\times$) often even reduces overall mAP. This can be explained by our augmentation strategy, which reuses the same background images and inserts only one synthetic vehicle per image. The re-use of the real images leads to limited new contextual diversity and inflates correlations between samples. Additionally, this approach does not reduce the strong class imbalance present in VisDrone1000 (27,447 cars vs. 417 buses, 1,036 trucks, and 2,192 vans). As a result, the model continues to see cars far more frequently than other classes, and excessive synthetic additions may lead to overfitting to repetitive patterns rather than improving generalization. Future work should explore strategies that synthesize multiple objects per image or introduce greater background diversity to mitigate these effects.

5.4 Synthetic data quality

No human-in-the-loop filtering, to remove unrealistic synthetic samples, was applied in this study. Although AnyDoor generally produces well-harmonized objects, occasional artifacts, such as distorted geometry, missing parts, or blurry textures, were observed. While these occurred infrequently (approximately 1 in 25 images for zero-shot and 1 in 50 for fine-tuned models), even a small number of unrealistic samples can negatively impact training by introducing noise and causing the detector to learn incorrect features. This highlights the potential value of incorporating human verification or automated quality control in future work. Although generative augmentation theoretically enables large-scale dataset expansion, this need for manual filtering of low-quality samples limits scalability. A possible solution would be to automate this process by applying thresholds on image-quality metrics such as the FID. However, it needs to be determined which metrics reliably capture inpainting quality.

5.5 Conclusion

This study demonstrates that diffusion-based inpainting shows potential as data augmentation strategy for UAV-based vehicle detection in low-resource scenarios. Zero-shot AnyDoor augmentation substantially improved detection for one of the minority classes. Under the tested conditions, fine-tuning provided limited additional benefit. Future research should focus on strategies to make fine-tuning more effective, particularly for the military domain where zero-shot performance could be weak. Another promising direction is to synthesize multiple objects per image to better address class imbalance and increase contextual diversity.

REFERENCES

[1] Du, D., Zhu, P., Wen, L., Bian, X., Lin, H., and et al., "VisDrone-DET2019: The vision meets drone object detection in image challenge results," in [*2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*], 213–226 (2019).

[2] van Leeuwen, M. C., Fokkinga, E. P., Huizinga, W., Baan, J., and Heslinga, F. G., "Toward versatile small object detection with Temporal-YOLOv8," *Sensors* **24**(22) (2024).

[3] Heslinga, F. G., Ruis, F., Ballan, L., van Leeuwen, M. C., Masini, B., van Woerden, J. E., den Hollander, R. J. M., Berndsen, M., Baan, J., Dijk, J., and Huizinga, W., "Leveraging temporal context in deep learning methodology for small object detection," in [*Artificial Intelligence for Security and Defence Applications*], **12742**, SPIE Sensors + Imaging (2023).

[4] Zhang, Y., Doughty, H., and Snoek, C. G., "Low-resource vision challenges for foundation models," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 21956–21966 (2024).

[5] Eker, T. A., Heslinga, F. G., Ballan, L., den Hollander, R. J., and Schutte, K., "The effect of simulation variety on a deep learning-based military vehicle detector," in [*Artificial Intelligence for Security and Defence Applications*], **12742**, 183–196, SPIE Sensors + Imaging (2023).

[6] Moate, C. P., Hayward, S. D., Ellis, J. S., Russell, L., Timmerman, R. O., Lane, R. O., and Strain, T. J., "Vehicle detection in infrared imagery using neural networks with synthetic training data," in [*Image Analysis and Recognition: 15th International Conference, ICIAR 2018, Póvoa de Varzim, Portugal, June 27–29, 2018, Proceedings 15*], 453–461, Springer (2018).

[7] Heslinga, F. G., Eker, T. A., Fokkinga, E. P., van Woerden, J. E., Ruis, F. A., den Hollander, R. J. M., and Schutte, K., "Combining simulated data, foundation models, and few real samples for training object detectors," in [*Synthetic Data for Artificial Intelligence and Machine Learning: Tools, Techniques, and Applications II*], **13035**, SPIE Defense + Commercial Sensing (2024).

[8] Ruis, F. A., Liezenga, A. M., Heslinga, F. G., Ballan, L., den Hollander, R. J., van Leeuwen, M. C., Masinia, B., Dijk, J., and Huizinga, W., "Improving object detector training on synthetic data by starting with a strong baseline methodology," in [*Synthetic Data for Artificial Intelligence and Machine Learning: Tools, Techniques, and Applications II*], **13035**, SPIE Defense + Commercial Sensing (2024).

[9] Fokkinga, E. P., te Hofsté, M. E., den Hollander, R. J., van der Meer, R., Benders, F. P., ter Haar, F. B., Marquis, V. E., van Berkel, M., Voogd, J. M., Eker, T. A., et al., "The validation of simulation for testing deep-learning-based object recognition," in [*Artificial Intelligence for Security and Defence Applications II*], **13206**, 253–275, SPIE (2024).

[10] Heslinga, F. G., Fokkinga, E. P., Eker, T. H., Liezenga, A. M., den Hollander, R. J. M., Oppeneer, V. O., van Heteren, A. M., van Vossen, R., Kuijf, H. J., van de Sande, J. J. M., van der Burg, D. W., Weyland, L. F., Henderson, H. C., Schadd, M. P. D., and Schutte, K., "On the use of simulated data for target recognition and mission planning," in [*Artificial Intelligence for Security and Defence Applications II*], **13206**, SPIE Sensors + Imaging (2024).

[11] Eker, T. A., Fokkinga, E. P., Heslinga, F. G., and Schutte, K., "Balancing 3D-model fidelity for training a vehicle detector on simulated data," in [*Artificial Intelligence for Security and Defence Applications II*], **13206**, SPIE Sensors + Imaging (2024).

[12] Rozantsev, A., Lepetit, V., and Fua, P., "On rendering synthetic images for training an object detector," *Computer Vision and Image Understanding* **137**, 24–37 (2015).

[13] Rojas, D. J. B., Fernandes, B. J. T., and Fernandes, S. M. M., "A review on image inpainting techniques and datasets," in [*2020 33rd SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*], 240–247, IEEE (2020).

[14] Jam, J., Kendrick, C., Walker, K., Drouard, V., Hsu, J. G.-S., and Yap, M. H., "A comprehensive review of past and present image inpainting methods," *Computer vision and image understanding* **203**, 103147 (2021).

[15] Fokkinga, E. P., Eker, T. A., van Woerden, J. E., Witon, J.-M., Stallinga, S. O. B., Visser, A., Schutte, K., and Heslinga, F. G., "Generative AI methods for synthesis of image data to train AI for automated scene understanding in a military context: a review of opportunities," in [*Synthetic Data for Artificial Intelligence and Machine Learning: Tools, Techniques, and Applications III*], **13459**, SPIE Defense + Commercial Sensing (2025).

[16] Chen, X., Huang, L., Liu, Y., Shen, Y., Zhao, D., and Zhao, H., "Anydoor: Zero-shot object-level image customization," in [*Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*], 6593–6602 (2024).

[17] Zhang, X., Guo, J., Yoo, P., Matsuo, Y., and Iwasawa, Y., "Paste, inpaint and harmonize via denoising: Subject-driven image editing with pre-trained diffusion model," *arXiv preprint arXiv:2306.07596* (2023).

[18] Yang, B., Gu, S., Zhang, B., Zhang, T., Chen, X., Sun, X., Chen, D., and Wen, F., "Paint by example: Exemplar-based image editing with diffusion models," (2022).

[19] Liu, H., Yang, H., Huijben, E. M. C., Schuiveling, M., Su, R., Pluim, J. P. W., and Veta, M., "PathoPainter: Augmenting histopathology segmentation via tumor-aware inpainting," *arXiv preprint arXiv:2503.04634* (2025).

[20] Hu, X. and Shi, Y., "Inpainting is all you need: A diffusion-based augmentation method for semi-supervised medical image segmentation," *arXiv preprint arXiv:2506.23038* (2025).

[21] Lyhs, J., Hinneburg, L., Fischer, M., Ölsner, F., Milz, S., Tschirner, J., and Mäder, P., "Bootstrapping corner cases: High-resolution inpainting for safety critical detect and avoid for automated flying," *arXiv preprint arXiv:2501.08142* (2025).

[22] Zhu, P., Wen, L., Bian, X., Ling, H., and Hu, Q., "Vision meets drones: A challenge," *arXiv preprint arXiv:1804.07437* (2018).

[23] Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J., "YOLOX: Exceeding YOLO series in 2021," *arXiv preprint arXiv:2107.08430* (2021).

[24] Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., and Fleet, D. J., "Synthetic data from diffusion models improves imagenet classification," *arXiv preprint arXiv:2304.08466* (2023).

[25] Mustikovela, S. K., De Mello, S., Prakash, A., Iqbal, U., Liu, S., Nguyen-Phuoc, T., Rother, C., and Kautz, J., "Self-supervised object detection via generative image synthesis," in [*Proceedings of the IEEE/CVF International Conference on Computer Vision*], 8609–8618 (2021).

[26] Ge, Y., Xu, J., Zhao, B. N., Joshi, N., Itti, L., and Vineet, V., "DALL-E for detection: Language-driven compositional image synthesis for object detection," *arXiv preprint arXiv:2206.09592* (2022).

[27] Liu, L., Muelly, M., Deng, J., Pfister, T., and Li, L.-J., "Generative modeling for small-data object detection," in [*Proceedings of the IEEE/CVF international conference on computer vision*], 6073–6081 (2019).
[28] Bie, F., Yang, Y., Zhou, Z., Ghanem, A., Zhang, M., Yao, Z., Wu, X., Holmes, C., Golnari, P., Clifton, D. A., He, Y., Tao, D., and Song, S. L., "Renaissance: A survey into ai text-to-image generation in the era of large model," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **47**(3), 2212–2231 (2025).
[29] Song, J., Meng, C., and Ermon, S., "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502* (2020).
[30] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B., "High-resolution image synthesis with latent diffusion models," *arXiv preprint arXiv:2112.10752* (2022).
[31] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative adversarial networks," *arXiv preprint arXiv:1406.2661* (2014).
[32] Dhariwal, P. and Nichol, A., "Diffusion models beat GANs on image synthesis," in [*Advances in Neural Information Processing Systems (NeurIPS)*], (2021).
[33] Kingma, D. P. and Welling, M., "Auto-encoding variational Bayes," *arXiv preprint arXiv:1312.6114* (2013).
[34] Gupta, A., Yu, L., Sohn, K., Gu, X., Hahn, M., Li, F.-F., Essa, I., Jiang, L., and Lezama, J., "Photorealistic video generation with diffusion models," in [*European Conference on Computer Vision*], 393–411 (2024).
[35] Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., and Bermano, A. H., "Human motion diffusion model," in [*International Conference on Learning Representations (ICLR)*], (2023).
[36] Yuan, Y., Song, J., Iqbal, U., Vahdat, A., and Kautz, J., "Physdiff: Physics-guided human motion diffusion model," in [*Proceedings of the IEEE/CVF international conference on computer vision*], 16010–16021 (2023).
[37] Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B., "Diffwave: A versatile diffusion model for audio synthesis," *arXiv preprint arXiv:2009.09761* (2020).
[38] Huang, R., Huang, J., Yang, D., Ren, Y., Liu, L., Li, M., Ye, Z., Liu, J., Yin, X., and Zhao, Z., "Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models," in [*Proceedings of the 40th International Conference on Machine Learning*], Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., eds., *Proceedings of Machine Learning Research* **202**, 13916–13932, PMLR (23–29 Jul 2023).
[39] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M., "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125* (2022).
[40] Labs, B. F., "Flux.1-dev." https://huggingface.co/black-forest-labs/FLUX.1-dev (2024). Accessed: 2025-03-17.
[41] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I., "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020* (2021).
[42] Zhang, L., Rao, A., and Agrawala, M., "Adding conditional control to text-to-image diffusion models," *arXiv preprint arXiv:2302.05543* (2023).
[43] Zhao, S., Chen, D., Chen, Y.-C., Bao, J., Hao, S., Yuan, L., and Wong, K.-Y. K., "Uni-controlnet: All-in-one control to text-to-image diffusion models," *arXiv preprint arXiv:2305.16322* (2023).
[44] Wang, Y., Xu, H., Zhang, X., Chen, Z., Sha, Z., Wang, Z., and Tu, Z., "OmniControlNet: Dual-stage integration for conditional image generation," *arXiv preprint arXiv:2406.05871* (2024).
[45] Bolanos, L., Urwin, G., Walsh, R., Clark, R., Hamari, J., and Zardadi, M., "EO2IR ControlNet: synthetic infrared image generation for automatic target recognition: experimental results in MIST," in [*Synthetic Data for Artificial Intelligence and Machine Learning: Tools, Techniques, and Applications III*], **13459**, SPIE Defense + Commercial Sensing (2025).
[46] Iizuka, S., Simo-Serra, E., and Ishikawa, H., "Globally and locally consistent image completion," *ACM Transactions on Graphics* **36**(4), 1–14 (2017).
[47] Li, Y., Liu, S., Yang, J., and Yang, M.-H., "Generative face completion," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 3911–3919 (2017).
[48] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S., "Generative image inpainting with contextual attention," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 5505–5514 (2018).

[49] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al., "Segment anything," in [*Proceedings of the IEEE/CVF international conference on computer vision*], 4015–4026 (2023).

[50] Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., and Cohen-Or, D., "An image is worth one word: Personalizing text-to-image generation using textual inversion," *arXiv preprint arXiv:2208.01618* (2022).

[51] Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K., "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," *arXiv preprint arXiv:2208.12242* (2022).

[52] Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al., "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193* (2023).

[53] He, R., Sun, S., Yu, X., Xue, C., Zhang, W., Torr, P., Bai, S., and Qi, X., "Is synthetic data from generative models ready for image recognition?," *arXiv preprint arXiv:2210.07574* (2022).

[54] Trabucco, B., Doherty, K., Gurinas, M., and Salakhutdinov, R., "Effective data augmentation with diffusion models," *arXiv preprint arXiv:2302.07944* (2023).

[55] Zhu, J., Li, S., Liu, Y., Huang, P., Shan, J., Ma, H., and Yuan, J., "ODGEN: Domain-specific object detection data generation with diffusion models," *arXiv preprint arXiv:2405.15199* (2024).

[56] Zhao, H., Sheng, D., Bao, J., Chen, D., Chen, D., Wen, F., Yuan, L., Liu, C., Zhou, W., Chu, Q., Zhang, W., and Yu, N., "X-paste: Revisiting scalable copy-paste for instance segmentation using CLIP and StableDiffusion," in [*Proceedings of the 40th International Conference on Machine Learning*], **202**, 42098–42109 (2023).

[57] Zhou, X., Koltun, V., and Krähenbühl, P., "Probabilistic two-stage detection," *arXiv preprint arXiv:2103.07461* (2021).

[58] Gupta, A., Dollar, P., and Girshick, R., "Lvis: A dataset for large vocabulary instance segmentation," in [*Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*], 5356–5364 (2019).

[59] Petersen, J., Abati, D., Habibian, A., and Wiggers, A., "Scene-aware location modeling for data augmentation in automotive object detection," *arXiv preprint arXiv:2504.17076* (2025).

[60] Ren, S., He, K., Girshick, R., and Sun, J., "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems* **28** (2015).

[61] Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., Darrell, T., et al., "BDD100K: A diverse driving video database with scalable annotation tooling," *arXiv preprint arXiv:1805.04687* **2**(5), 6 (2018).

[62] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P., "Microsoft COCO: Common objects in context," (2015).

[63] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M., "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934* (2020).

[64] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D., "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412* (2017).