Generating Synthetic Datasets to Train Deep Models for Counting in Large-Scale Animal Groups

Iroda Ibrohimova, Madina Mirzatayeva, Gianni A. Di Caro

Carnegie Mellon University in Qatar, Doha, Qatar

ABSTRACT

Accurate animal counting plays a critical role in wildlife monitoring tasks. In scenarios involving large animal aggregations, manual counting from images or videos is often impractical. Although recent deep learning and machine vision techniques offer promising automated solutions, their effectiveness and generalization across different species and scenarios are limited by the scarcity of large, diverse, and well-annotated datasets that capture the complexity of dense animal groups across varying species, viewpoints, and conditions. This work addresses this limitation by developing a process pipeline for the automatic and extensive generation of large-scale synthetic datasets able to capture a wide variety of conditions and scenarios. We use Blender to create realistic 3D animal models and we adapted Blender's Boid Particle System to create realistic 3D video animations of animal group motion. We have considered fish, birds, and mammals groups. The synthetic video generation freely allows variation in camera views and animal densities, and to add precise annotations of each image frame. We have used the synthetic datasets to fine-tune state-of-the-art deep learning models for repeated object counting. Experiments show that both the YOLO and CSRNet models used as a reference can significantly boost their counting and detection performance, as well as their generalization capabilities across different scenarios.

Keywords: Animal counting, synthetic datasets, Boids simulation, 3D animation, deep learning.

1. INTRODUCTION

Accurate *animal counting* is essential for wildlife monitoring, conservation, and ecosystem management. In the presence of large aggregations, manual counting from image or video collections (e.g., from fixed or mobile cameras) is often impractical or even unfeasible. Artificial intelligence, deep learning, and machine vision techniques have shown great potential to automate counting tasks in images, such as counting objects, people, and animals (e.g.[1–6]). Typically, current reference models for counting in images such as the various YOLO[1] models, struggle with multiple occlusions, generalization across species, consistent performance across camera viewpoints, and dense aggregations like bird flocks or fish schools.[7–9] A general key barrier is the *scarcity of large annotated datasets* for supervised training that effectively cover this diversity in terms of animal aggregations, species (morphology, sizes, colors, etc.), and viewpoints. For instance, a model trained on aerial bird's-eye views of mammal herds from drones may not generalize to images from a canopy-level sensor. Unfortunately, extensive and precise manual annotation is both time-consuming and difficult, especially in high-density scenarios.[10]

To address data scarcity, we developed a synthetic data generation pipeline that scalably produces annotated datasets across the wide range of conditions needed for training or fine-tuning deep learning models for animal counting. Using Blender, we generate realistic 3D models of animals and environments, and tune Blender's Boids Particle System[11] to simulate realistic group motion for fish, birds, and terrestrial mammals. The pipeline outputs high-resolution video animations where the same sequence can be viewed under multiple viewpoints and lighting. Based on viewpoint and occlusions, each frame is automatically annotated for counting, detection bounding boxes, and occlusion levels. Thus, the datasets support both density map–based models (e.g., CSRNet[4]) and bounding box–based models (e.g., YOLO[1]).

Corresponding author: Gianni A. Di Caro

E-mail: {iibrohim, mmirzata, gdicaro}@andrew.cmu.edu

Madina Mirzatayeva and Iroda Ibrohimova have equally contributed to the work. They are both undergraduate students.

We used synthetic datasets to fine-tune pre-trained deep learning models, in particular YOLOv8x[12] and CSRNet[4] used as references.Results show that synthetic data effectively boosts animal counting performance across densities and species. For fish, YOLOv8x and CSRNet trained on synthetic data achieved notable gains in accuracy and generalization in both low- and high-density scenes. For mammals, YOLOv8x fine-tuned with synthetic zebra data improved performance, and when trained only on synthetic deer data, generalized effectively to real images despite no real examples in training. These findings highlight the critical role of synthetic data in enhancing model robustness when annotated datasets are scarce.

2. RELATED WORK

Deep learning (DL) has transformed wildlife monitoring by enabling automated detection and counting of animals using imagery from drones, satellites, thermal cameras, and camera traps. Compared to traditional manual approaches, DL object detection models like YOLO, Faster R-CNN,[2] and U-Net,[3] as well as density estimation models such as CSRNet, offer improved scalability and accuracy across a wide range of species and environments.[8,9,13] These methods have been successfully applied to fish, mammals, and birds supporting both conservation and ecological research in accessible and remote settings.[10]

However, three key limitations restrict the scope and robustness of broader adoption. First, these models are highly dependent on large, diverse, and well-annotated datasets, which are often unavailable, especially in scenes with dense animal aggregations and visually complex conditions, such as occlusion, motion blur, and varied perspectives.[7,8] Second, accurate counting becomes difficult when individuals overlap in tight groups, as seen in flocks, herds, or schools.[9] Third, current models often generalize poorly across species, habitats, and sensing conditions, limiting their long-term applicability in ecological monitoring.[14]

To overcome data scarcity and complexity in animal detection, synthetic datasets have been explored for their scalability and control over factors like occlusion, lighting, and density. Smith et al.[15] showed that combining real and synthetic images improved YOLOv9 accuracy in poultry detection, while Zhang et al.[16] enhanced chicken facial detection using synthetic augmentation. Tools like Blender, Unity, and Unreal Engine, along with domain randomization,[17] have been used to improve model robustness. However, such efforts remain largely limited to low-density scenarios like poultry, with little application to densely packed mammals or fish.

Fish schools and mammal herds present unique challenges due to dynamic motion, occlusion, and dense formations. In aquatic settings, models like AquaYOLO and FishDet-YOLO[18,19] adapt YOLO architectures for underwater imagery, but training datasets often lack sufficient object density, e.g., OzFish averages 25 fish per image, while others fall below one.[20] YOLO-based detectors tend to undercount in such cases,[21] whereas density map-based models like CSRNet[4] have shown improved accuracy by generating continuous maps.[22] For mammals, YOLO[1] variants have been applied across varied environments: red foxes,[23] forest scenes,[24] and thermal imagery.[25,26] Attention-based improvements further enhanced detection in large species like elephants.[27] Still, high-density counting remains underexplored; methods like YOLO-SDD,[28] designed for poultry occlusion, have not been widely extended to mammalian groups.

To address these gaps, we propose a modular synthetic data framework tailored to dense animal aggregations. Exploiting and tweaking Blender and its Boid Particle System, we simulate 3D realistic animal group behaviors with automatic annotation and environmental variations. Our datasets provide higher per-image density, occlusion-aware labeling, and support for both detection and density models, offering a scalable solution for underrepresented animal counting scenarios.

3. METHODS

3.1 Animal Group Simulation with Blender

We have used Blender to create visually realistic animal models, and its Boids Particle System to simulate realistic animal group behaviors. The Boids engine in Blender is built on the Reynolds Boids model[11] and can be used to effectively mimic general collective motion using simple rules: separation, alignment, and cohesion. We have used it to control the collective behavior of birds, fish, and mammals. We customized motion for each species (e.g., 3D freeform for fish and birds, and ground-constrained planar motion for deer and zebra) using Blender's physics and particle emitters.

We selected Blender over other engines due to its high-fidelity rendering, scripting support via Python, and ease of automating dataset generation. Detailed animal and environmental models were imported from Blender's built-in libraries like Sketchfab and BlenderKit, and the scenes were scripted to control agent dynamics, camera movements, and rendering.

3.2 Automated Labeling for Object Counting and Density Map-based methods

We aimed to create annotated datasets that can be widely used by DL methods for animal counting. Roughly speaking, these methods can be categorized in object detection models (e.g., YOLO), requiring bounding boxes to annotate object location and size, and density map-based counting models (e.g., CSRNet), which only use object centroids. Therefore, we created automated object and animal labeling compliant with both methods.

In particular, using Blender's scripting interface, we automated object labeling through camera projections. The 3D bounding box of each visible object was projected into 2D using the `world_to_camera_view` function. These were used to compute YOLO-compatible normalized bounding box, center, and size. Adjustments were made to account for coordinate system differences between Blender and YOLO (e.g., y-axis flipping). Each rendered frame (PNG) was saved alongside a corresponding YOLO annotation text file. We supported both static and dynamic camera modes to simulate varied visual perspectives, including UAV-like motion. Figure 1 shows examples of rendered animals and bounding boxes for different species of animals.

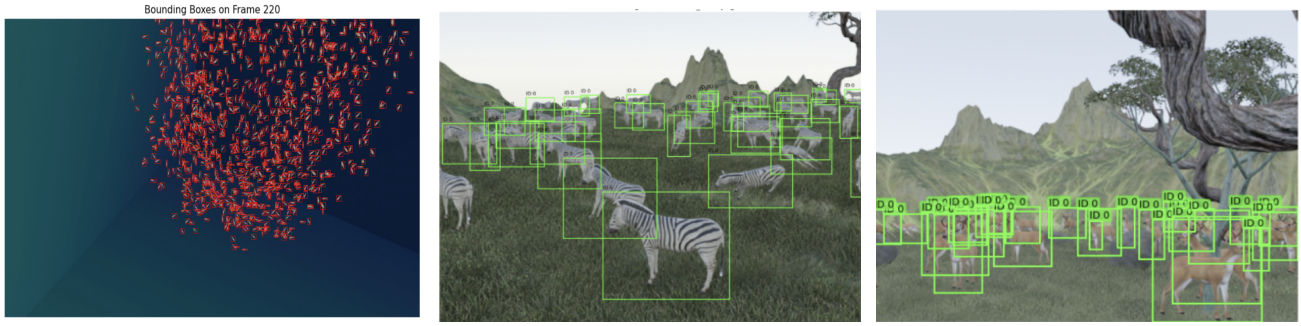

Figure 1. Examples of YOLO training data showing fish, zebra, and deer images with bounding box annotations.

To create datasets for density map-based approaches, we converted YOLO's bounding boxes into point annotations by extracting the center of each box. These were saved as `.npy` files with shape `(N, 2)`. Each pair of image points was used to generate density maps by convolving points with Gaussian kernels ($\sigma = 15$). Heatmaps were created dynamically during training and images were resized to 384×512, with density maps downsampled to 48×64. Figure 2 shows an example of this process for fish schools.

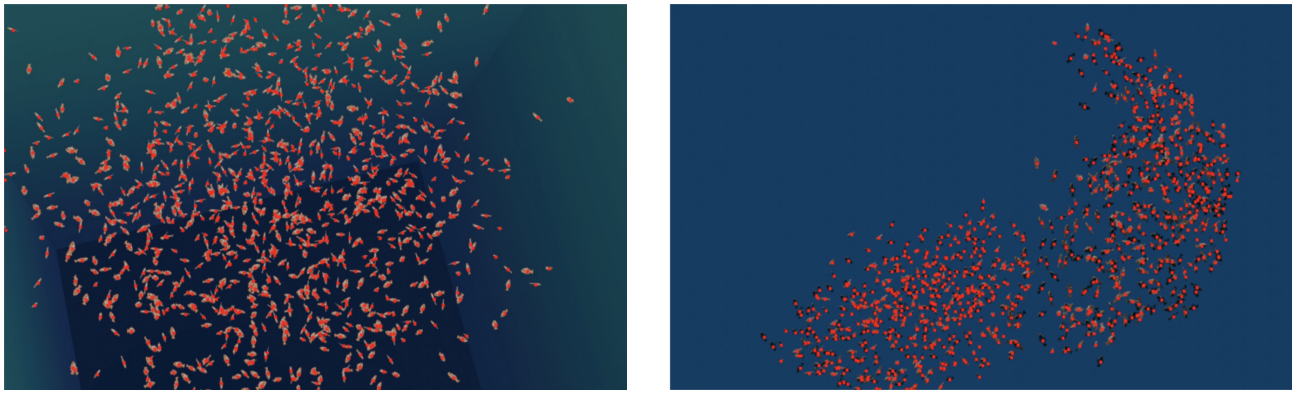

Figure 2. Heatmap outputs of fish density generated for the density map-based model training (e.g., CSRNet).

3.3 Post-Processing and Data Cleaning

We addressed issues from occlusion and projection artifacts by filtering out overlapping or invisible animals. First, duplicate 2D coordinates were removed. Second, we computed object-level occlusion using intersection-over-union (IoU), which measures the overlap between a predicted and ground truth bounding box as the ratio of their intersection to their union. We discarded detections with over 90% occlusion. Third, we made a y-axis flip to correct Blender's coordinate misalignment. The early frames (first 100-300) from each simulation, which included swarm formation artifacts from Blender, were excluded to improve the quality of the data set.

3.4 Datasets for animal counting

We created separate datasets for the two mentioned classes of DL, object detection and density map-based counting. We have considered aquatic, aerial, and terrestrial environments. For fish, we generated 12,000 high-density images (400–1400 fish per frame), as well as lower-density datasets of 500, 1,000, and 5,000 images (0–400 fish per frame). For birds, we generated approximately 15,000 images with 0–1,200 individuals, captured from top, side, and oblique angles to simulate realistic aerial views. For mammals, we generated two distinct datasets: one for zebras (5710 images) and one for deer (14600 images).

Together, the datasets support comparative analysis of detection and counting methods across different species, scene complexities, variations in density, as well as dataset sizes.

4. EXPERIMENTAL ANALYSIS

To validate the quality and usefulness of the synthetically generated animal datasets, we have used them to fine-tune pre-trained models, more precisely, YOLOv8x[12] and CSRNet,[4] taken as representatives of object detection and density map-based approaches, respectively.

YOLO reformulates object detection as a single regression problem. The image is divided into an $S \times S$ grid; each grid cell predicts B bounding boxes and class probabilities A prediction includes coordinates (x, y, w, h) and a confidence score. As test score, YOLO reports the Mean Average Precision, mAPx, which is a measure of how accurately a model detects and correctly localizes objects, where x indicates how closely the predicted boxes need to match the true object locations for the detection to count as correct. E.g., mpA50 measures how well the model detects objects with at least 50% overlap between predicted and true boxes, averaged over all classes. YOLO is expected to be effective for relatively sparse scenes, but may fail to be accurate in dense scenes.

CSRNet, in contrast, is a density map-based regression model designed for high-density scenes.[19] It predicts a density map $D(x, y)$ over an image of width W and height H, and the total object count is $\hat{C} = \sum_{x=1}^{W} \sum_{y=1}^{H} D(x, y)$. From point-annotated data, ground-truth maps are generated by placing Gaussian kernels at each point. The model is trained using the MSE loss between predicted and ground truth maps.

We have designed a set of experiments to assess whether models trained solely on synthetic data can achieve reliable performance on both real and synthetic data, and whether they can generalize across different animal species and observation conditions, thereby demonstrating the practical value of the generated datasets.

4.1 Experimental Setup

We evaluated object counting using synthetic datasets simulating dense animal aggregations, with fish (aquatic) and mammals (terrestrial) under varied occlusion and density. Note that data about the bird dataset are not reported because of space constraints. Models were assessed using MAE, RMSE, and MAPE measures of error (below, y_i is the count output by a model, $\hat{y}$ is the true count).

$$\text{MAE} = \frac{1}{n} \sum |y_i - \hat{y}_i|, \quad \text{RMSE} = \sqrt{\frac{1}{n} \sum (y_i - \hat{y}_i)^2}, \quad \text{MAPE} = \frac{100}{n} \sum \left| \frac{y_i - \hat{y}_i}{y_i + \epsilon} \right|$$

During training, we monitored box loss (bounding box accuracy), class loss (object classification accuracy), and Distribution Focal Loss (DFL), which refines localization by focusing on box center regions. These losses guide the model in learning to detect, classify, and localize animals effectively, while the curves help evaluate performance under varied visual complexity.

To better assess detection behavior, we analyzed also three diagnostic curves: Recall–Confidence (RC), Precision–Recall (PR), and Precision–Confidence (PC). These reveal how recall, precision, and prediction reliability evolve with confidence thresholds, especially important in dense aggregations.

YOLOv8x was trained for 70 epochs using the Adam optimizer (initial learning rate 0.001) with cosine decay. A batch size of 16 and input resolution of 640×640 were used, with data augmentation and Automatic Mixed Precision (AMP) enabled. YOLO minimizes a loss that combines localization, confidence, and classification errors to train object detectors that predict accurate bounding boxes, object presence, and class labels.

The pretrained CSRNet combines a VGG16 frontend with a custom dilated convolution backend. It was trained for 50 epochs using Adam (learning rate 1×10^{-4}) with a batch size of 128. Input images were resized to 384×512 and density maps to 48×64. Gaussian smoothing kernels with $\sigma = 15$ was applied to annotations. The loss is a pixel-wise mean squared error that penalizes differences between the predicted and ground truth density maps across all image pixels.

4.2 Fish Counting with YOLOv8x and CSRNet

YOLOv8x was trained on 5,000 synthetic fish images with counts between 100 and 400. CSRNet was trained on 12,000 high-density images with counts between 400 and 1,400. Both models were tested separately on 1,000 test images from synthetic dataset and six real-world samples, which we manually annotated (note that annotated datasets for dense fish counting do not seem to be available on the Internet).

According to the training dynamics of YOLOv8x on synthetic fish aggregation data, the training losses (box, class, and DFL) consistently decreased, with rapid convergence in the first 40 epochs and stabilization after 50. Validation losses followed similar trends, confirming generalization across unseen samples. Notably, mAP50 reached 0.97, and mAP50–95 improved steadily, reflecting robustness across IoU thresholds.

The model's inference performance was assessed using the RC, PR, PC curves. The RC curve indicates strong recall at most confidence thresholds, with only a slight decline above 0.85. The PR curve demonstrates excellent separability, achieving AP of 0.967 for the fish class. Precision increased with confidence, peaking at 1.00 around a threshold of 0.97, suggesting high-quality predictions at higher confidence.

To evaluate counting accuracy, MAE, RMSE, and MAPE trends were calculated on 1,000 test images. MAE remained below 25, even in denser scenes. RMSE peaked slightly for complex aggregations, but remained below 40. MAPE varied between 1%–10%, highlighting the consistency on different count scales. These results show YOLOv8x's capability in object localization and count estimation for moderately dense aquatic environments.

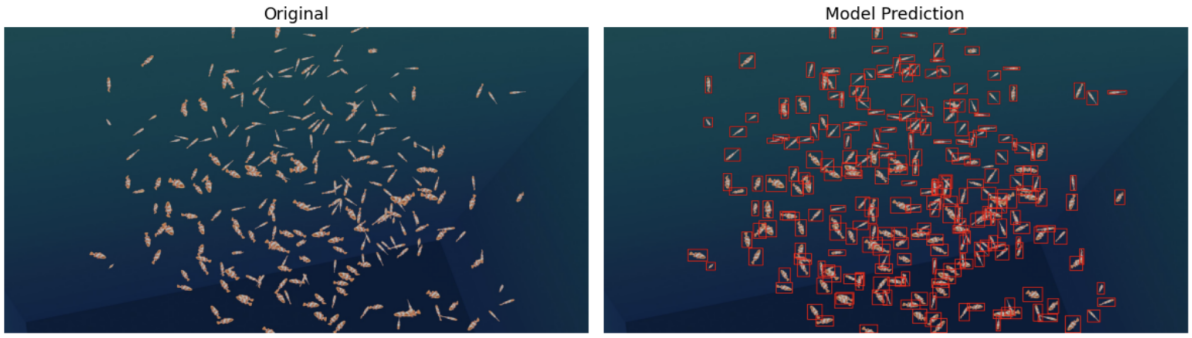

Figure 3. YOLOv8x model predictions with bounding boxes on previously unseen fish aggregation images

We evaluated YOLOv8x on previously unseen real-world images (see Figure 4) and found that the fine-tuned model accurately approximated fish counts, underestimating around 50 in high-density sample. Instead, the pretrained model failed to track count trends, underscoring the value of synthetic data for domain adaptation. In addition, we tested YOLOv8x models trained on synthetic datasets of 500, 1,000, and 5,000 images. MAE decreased from 46 (500 images) to 7.5 (5,000 images), and RMSE dropped from 46 to 9.1, confirming that increasing synthetic data volume significantly improves detection performance.

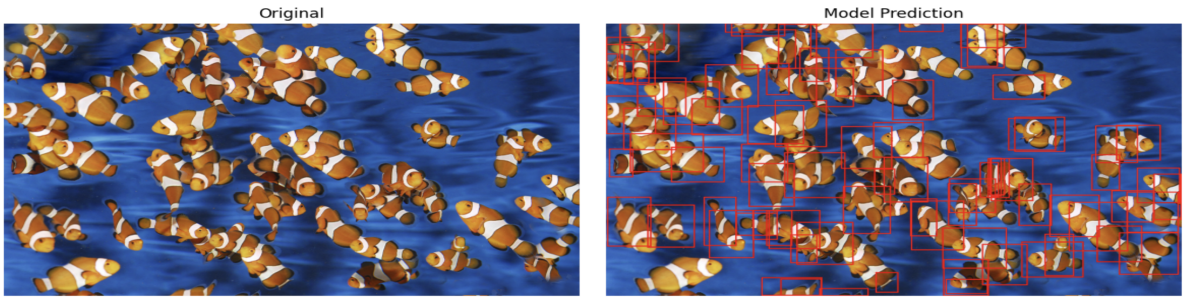

Figure 4. Example of a real-life image used to test the fine-tune YOLOv8x model.

4.2.1 Performance Evaluation of CSRNet for Fish Counting

CSRNet was evaluated on high-density fish scenes. The model achieved MAE of 23.12, RMSE of 30.04, and MAPE of 2.77%, calculated by comparing total predicted count (sum of density map) with ground truth.

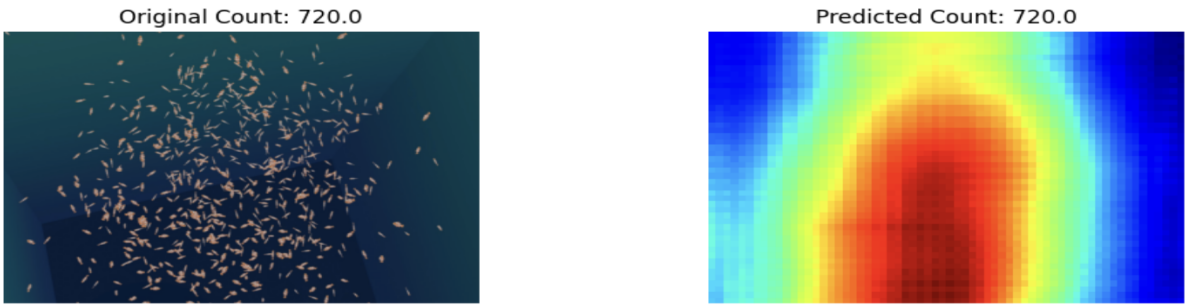

Figure 5. Predicted fish counts and CSRNet-generated density heatmap for a test image.

Visual results (Figure 5) show CSRNet accurately capturing fish concentration patterns in dense scenes. The predicted heat maps correlate closely with the true distributions of the fish. Most per-image error counts remained below 25, with few exceeding 50. These results confirm CSRNet's suitability for density estimation in heavily occluded environments. The model generalizes well across crowd sizes, demonstrating precise regression of total counts and robustness in extreme aggregation conditions.

4.3 Mammal Detection and Counting with YOLOv8x

Environment, appearance, and aggregation patterns of terrestrial mammals are different from fish. To evaluate the impact of synthetic datasets on detection mammals, we trained two YOLOv8x models: one for the zebras and one for the deer. YOLO is typically pretrained on the COCO dataset, a widely used benchmark containing labeled images of 80 common object categories, including zebra. While COCO includes some deer-like species (such as elk or moose), it does not contain a broad or consistent representation of deer. This setup let us examine two scenarios: using synthetic data to improve detection performance for a species already present in the pretrained model (zebra), and training a detector from scratch for a novel or underrepresented species (deer).

4.3.1 Zebra Model Experiments

The zebras dataset consists of 5,710 synthetic images, simulating herds from a few to 300+ zebras, using over 15 camera angles and 20% image augmentations. For benchmarking, we compiled 1,751 real images from Roboflow.[29] After manual filtering, we curated 1084 real images for training, 571 for validation, and selected 96 high-density real images for testing. Table 6 summarizes the experimental results. From all of the experiments, we can see that the best results were obtained when a mixture of synthetic and real images was used for both training and validation. The testing set included densely packed images, each with more than six zebras on average, making accurate counting challenging even for the human eye, yet the model trained on the mixed dataset outperformed others across all metrics. It also had good training with mAP scores of 0.982 (mAP@0.5) and 0.95 (mAP@0.5:0.95). Additionally, loss curves showed steady convergence with minimal overfitting. Figure 7 shows example images and predictions.

Model Training with zebra datasets				Performance metrics results		
Model	**Trained on**	**Validated on**	**Tested on**	**MAE**	**RMSE**	**MAPE**
YOLOv8x	COCO (default)	-	Real (96 images)	1.854	3.122	25.31%
YOLOv8x	Synthetic only (5713 images)	Real (1655 images)	Real (96 images)	5.396	6.875	69.00%
YOLOv8x	Mixed (Synthetic (4608) + Real (1084))	Mixed (Synthetic (1105) + Real (570))	Real (96 images)	1.438	2.327	17.93%
YOLOv8x	Real only (1084 images)	Real only (570 images)	Real (96 images)	1.812	2.428	24.15%

Figure 6. Model performance comparisons against 96 testing images

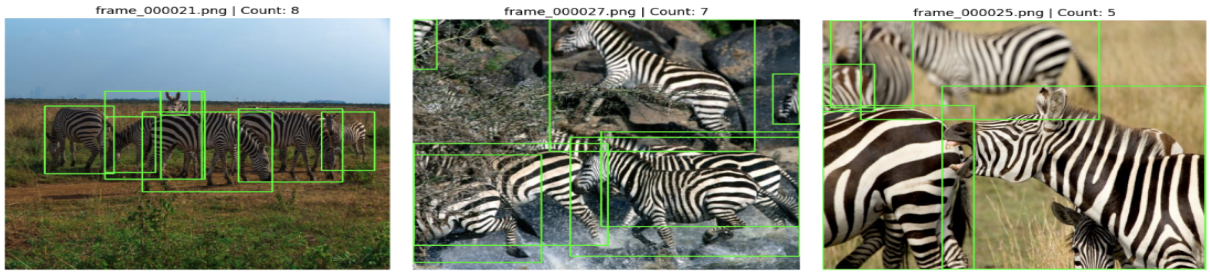

Figure 7. YOLOv8x trained on mixed dataset predictions on real zebra images

4.3.2 Deer Model Experiments

We generated 14,600 labeled synthetic deer images with added lighting variation in 2,488 frames to improve robustness. No large labeled deer datasets exist, especially for group formations, so we compiled 20 real images of deer herds for limited yet quantitative testing. Despite being trained solely on synthetic data and limited to a single deer species, our fine-tuned YOLOv8x model consistently outperformed the baseline YOLOv8x (pretrained on COCO) across all 20 testing images. As shown in Figure 8, the bottom row demonstrates our model's ability to correctly identify the majority of deer in diverse real-world environments, whereas the baseline model (top row) failed to detect deer in most cases.

Figure 8. Top: Baseline COCO-pretrained YOLOv8x on real deer images. Bottom: YOLOv8x trained on synthetic deer dataset tested on real images

Although we lacked a sufficiently labeled dataset to compute quantitative metrics such as MAE or RMSE, the qualitative results clearly demonstrate the superior generalization and detection capabilities of the model fine-tuned on synthetic data. Based on earlier zebra experiments, where even a modest amount of real data significantly improved accuracy, it is reasonable to conclude that combining synthetic and real datasets could further enhance performance. Overall, these findings support the effectiveness of synthetic data for enabling animal detection and counting in large aggregations, especially for species underrepresented in existing datasets.

5. CONCLUSIONS AND FUTURE WORK

This study introduced a scalable synthetic data generation pipeline using Blender and its Boids Particle System to realistically simulate animal aggregations under diverse environmental conditions, densities, and viewpoints. Richly annotated datasets were generated for fish, zebras, deer, and birds, and used to fine-tune YOLOv8x (detection-based) and CSRNet (density map-based), used as reference DL models for animal counting.

We ran experiments quantifying how synthetic datasets enhance pre-trained models in animal counting tasks. For low to mild dense fish aggregations (up to 400 fish per image), YOLOv8x trained on 500 synthetic images yielded a MAE of 46, while with 5,000 images MAE dropped to 7.5, showing that larger synthetic datasets significantly improve accuracy and generalization. For higher-density fish aggregations (400–1,400 per image), CSRNet trained on 12,000 synthetic images achieved a MAPE of just 2.77%. For mammals, YOLOv8x pre-trained on COCO dataset already performed well on zebras but still showed reduced MAE (1.854 to 1.438) when a mix of synthetic and few real samples was used. For deer, absent in COCO, models trained exclusively on synthetic data generalized effectively to all 20 real test images, despite no real deer data used in training. These results highlight that synthetic datasets not only can address data scarcity but can also effectively improve robustness across densities, species, and environments.

While our pipeline enables rapid and reproducible dataset generation across species and scenes, limitations remain. Models trained solely on synthetic data sometimes struggled with generalization in real images containing domain-specific clutter. Simulated behaviors were constrained by available 3D assets and animation logic, limiting interaction realism. These findings emphasize the value of even small real datasets and the need for validation.

Future work includes streamlining the pipeline with better automation in dataset generation, annotation, and quality control to reduce manual effort and support larger-scale training. Moving beyond static image analysis to video-based counting with spatiotemporal or recurrent architectures may further improve accuracy in dynamic scenes. Enhancing behavioral realism through more advanced animations (e.g., grazing, turning, interacting) would make synthetic data more representative and improve generalization to complex ecological environments.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," in *Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 91–99, 2015.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Int. Conf. on Medical Image Comp. & Comp-Assisted Inter. (MICCAI)*, pp. 234–241, 2015.

[4] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Rec. (CVPR)*, 2018.

[5] O. Urbann and J. Stenzel, "A convolutional neural network that self-contained counts," *Journal of Image and Graphics* **7**(4), pp. 112–116, 2019.

[6] A. M. Algamdi and H. M. Alghamdi, "Instant counting & vehicle detection during Hajj using drones," *Journal of Image and Graphics* **11**(2), pp. 204–211, 2023.

[7] J. Charco, A. Sappa, B. Vintimilla, and H. Velesaca, "Transfer learning from synthetic data in the camera pose estimation problem," in *Proc. of 15th VISIGRAPP, Vol. 4: VISAPP*, pp. 498–505, 2020.

[8] L. Xu, M. García, and G. Sánchez, "Deep learning for wildlife: Challenges of occlusion and perspective in aerial footage," *Eco-Informatics* **8**(3), pp. 45–62, 2024.

[9] Z. Xu, T. Wang, A. Skidmore, and R. Lamprey, "A review of deep learning techniques for detecting animals in aerial and satellite images," *Int. J. of Appl. Earth Observation and Geoinformation* **128**, p. 103732, 2024.

[10] A. Lamba, P. Cassey, R. R. Segaran, and L. P. Koh, "Deep learning for environmental conservation," *Biological Conservation* **257**, p. 109136, 2021.

[11] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH Computer Graphics* **21**(4), pp. 25–34, 1987.

[12] G. Jocher, A. Chaurasia, T. Qiu, and Ultralytics, "YOLOv8: Ultralytics next-generation object detection and segmentation model." https://github.com/ultralytics/ultralytics, 2023.

[13] S. Christin, E. Hervet, and N. Lecomte, "Applications for deep learning in ecology," 05 2018.

[14] B. G. Weinstein, "A computer vision for animal ecology," *J. of Animal Ecology* **87**(3), pp. 533–545, 2018.

[15] S. Cakic, T. Popovic, S. Krco, I. Jovovic, and D. Babic, "Evaluating the FLUX.1 synthetic data on YOLOv9 for ai-powered poultry farming," *Applied Sciences* **15**, p. 3663, Mar. 2025.

[16] X. Ma, X. Lu, Y. Huang, X. Yang, Z. Xu, G. Mo, Y. Ren, and L. Li, "An advanced chicken face detection network based on GAN and MAE," *Animals* **12**, p. 3055, Nov. 2022.

[17] J. Tremblay, A. Prakash, D. Acuna, *et al.*, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," *arXiv preprint* , 2018.

[18] K. S. Patro, V. Bharti, A. Sharma, A. Sharma, and V. Yadav, "Fish detection in underwater environments using deep learning," *National Academy Science Letters* , 05 2023.

[19] L. Yan, L. Zhang, X. Zheng, *et al.*, "Deep feature network with multi-scale fusion for highly congested crowd counting," *International Journal of Machine Learning and Cybernetics* **15**(3), pp. 819–835121078, 2024.

[20] S. H. Bengtson *et al.*, "Autofish: Dataset and benchmark for fine-grained analysis of fish," in *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pp. 1598–1607, 2025.

[21] S. Gong, Z. Ma, and X. Li, "Umfnet: Frequency-guided multi-scale fusion with dynamic noise suppression for robust low-light object detection," *Applied Sciences* **15**(10), p. 5362, 2025.

[22] D. Zhou, H. Tan, Y. Li, Y. Deng, and M. Zhu, "Line-labelling enhanced CNNs for transparent juvenile fish crowd counting," *Smart Agricultural Technology* , p. 100963, 2025.

[23] A. K. Schütz *et al.*, "Application of YOLOv4 for detection and motion monitoring of red foxes," *Animals* **11**, p. 1723, June 2021. Special Issue on *Animal Activity in Farms*.

[24] Z. Ma, Y. Dong, Y. Xia, D. Xu, F. Xu, and F. Chen, "Wildlife real-time detection in complex forest scenes based on YOLOv5s deep learning network," *Remote Sensing* **16**(8), p. 1350, 2024.

[25] P. Povlsen, D. Bruhn, P. Durdevic, D. O. Arroyo, and C. Pertoldi, "Using YOLO object detection to identify hare and roe deer in thermal aerial video footage," *Drones* **8**(1), p. 2, 2023.

[26] M. E. Çimen, "Comparison of deep learning and YOLOv8 models for fox detection around the henhouse," *Journal of Smart Systems Research* **5**(2), pp. 76–90, 2024.

[27] R. K. Mullick and R. K. Mandal, "An efficient elephant detection strategy using Visual Attention Network (VAN) in custom dataset improved YOLOv7 model," in *CEUR Workshop Proc.*, **3900**(32), p. 25, 2024.

[28] Y. Guo, Z. Wu, B. You, L. Chen, J. Zhao, and X. Li, "YOLO-SDD: An effective single-class detection method for dense livestock production," *Animals* **15**, p. 1205, Apr. 2025.

[29] B. Dwyer, J. Nelson, T. Hansen, *et al.*, "Roboflow." Computer Vision Software, 2025. Version 1.0.