Automatic Waterfowl and Habitat Detection using Drone Imagery and Deep Learning

Andrew Zhao
Computer Science Department
University of Illinois, USA

Andrea Fratila, Yang Zhang, Zhenduo Zhai, Zhiguang Liu, Yi Shang
Department of Electrical Engineering and Computer Science
University of Missouri, USA

2024 IEEE International Conference on Consumer Electronics (ICCE) | 979-8-3503-2413-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCE59016.2024.10444338

***Abstract*—The integration of drone technology and Artificial Intelligence (AI) has opened up new possibilities for wildlife conservation and habitat monitoring. In this paper, we present a new system for efficiently and accurately analyzing waterfowl populations and classifying their habitats over large natural areas using drone imagery and deep learning (DL). Given a sequence of drone images captured by a drone along a flight path, the system utilizes customized deep learning models for waterfowl detection and counting, Meta's SAM for image segmentation and customized deep learning models for segment classification, and ChatGPT to generate text-based survey reports. Several image overlap detection methods were developed and compared with. Our experimental results show accurate waterfowl and habitat detection results and improvement over previous work, providing efficient and accurate data analysis for wildlife conservation efforts.**

I. Introduction

The increasing availability of new technology, especially drones, has opened up new possibilities for aerial imaging and analysis in various domains, including wildlife conservation and habitat monitoring. In this work, we aim to leverage drone imagery and AI techniques to efficiently and accurately analyze waterfowl populations, classify their habitats, and generate informative reports over large natural areas. We develop an integrated system that utilizes various deep learning models to accurately count waterfowl and recognize their natural habitat seen on a drone flight, then finally generates a text-based report summarizing the gathered data.

Traditionally, bird population surveys and habitat analysis have relied on manual methods like field observation. These are time-consuming, labor-intensive, and limited in spatial coverage. By using drone imagery and AI techniques, we can overcome these limitations and achieve much more efficient data analysis while maintaining human-level accuracy. While others have already leveraged new technology to try and tackle this problem, the proposed system offers several advantages over previous works. By automating bird detection and classification using deep learning, we can vastly improve the speed in identifying waterfowl populations, especially with the large area coverage provided by a sequence of drone images. We also use image segmentation, classification, and text reports to enable users to obtain a comprehensive visualization of the study area. Finally, by detecting overlapping regions between consecutive images across a drone flight, we are able to avoid the issue of double-counting waterfowl in order to improve the overall counting accuracy. The system architecture is shown in Fig. 1.

The contributions of this paper are as follows:

- Develop and compare several new methods for detecting and removing overlapping regions between consecutive images captured by a drone along a flight path.
- Use Meta's open-source Segment Anything Model (SAM) [1] to identify and delineate distinct habitat regions within the images.
- Use image classification to determine the class associated with the generated masks from SAM.
- Integrate OpenAI's ChatGPT to generate a text-based report summarizing the AI-generated data, including bird population statistics, habitat classifications, and any other relevant data.

II. Related Work

The proposed work has several distinct parts: the integration of bird detection in drone imagery, image segmentation, habitat classification, image overlap detection, and report generation for environmental monitoring and analysis. In recent years, there have been several studies on these topics and other projects aiming to achieve similar goals regarding AI in conservation. Here is an overview of the relevant literature in each of these areas.

Of the existing wildlife detection programs using aerial imagery collected by UAVs as of 2016, birds have been used as subjects in 19 studies [2]. Waterfowls are the most popular subject amongst the birds, with all but one study using them as the subject. Before 2016, deep learning and ML were each used for automatic bird detection only once.

With the advancements of technology since, there are now numerous studies employing these methods, with one such similar study being [3], who have developed a model that has the ability to classify detected waterfowl in aerial images in addition to detection. The deep learning model that was used in our system was the general model created by [4], which only does detection but is more suited to our needs as it was trained on a large dataset similar to the one we use for this research [5].

Regarding waterfowl habitat classification, [6] use a traditional CNN model to segment drone images of bog vegetation in combination with a pixel-based Random Forest (RF) classifier to label their species. [7] classifies habitats on coastal

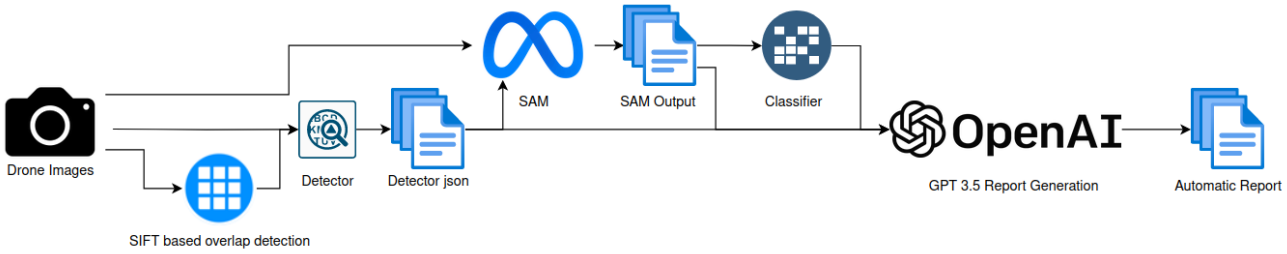

Fig. 1: The system architecture of the proposed pipeline.

regions using a pixel-based Support Vector Machine (SVM) classifier on drone images. Both these methods achieve a high accuracy of 85-90%.

In our pipeline, we use SAM [1], which through its revolutionary promptable segmentation can achieve better performance. Even though SAM is not intended for aerial images, the massive training dataset should allow the model to generalize to drone images and still perform well. However, SAM is not without pitfalls, as discussed by [8]. They found that SAM performs especially poorly on images with low contrast and when attempting to segment small regions. SAM's robustness was investigated in [9], where the model was fed with different user prompts and varying datasets, finding that SAM performs much worse on perturbed images compared to raw images and also that good user prompts can cause significant improvements in accuracy.

Regarding detecting the overlapping areas across consecutive drone images, [10] was the only paper that we were able to find related to this subject. In this paper, we propose several methods and the best method achieved an overlap prediction accuracy of roughly 0.85 on a test dataset of over 28,000 image pairs.

There are few studies that look into the quality of ChatGPT's data-to-text generation. [11] is one such study, where GPT-3.5 and GPT-4 were both used to generate text reports from radiology reports using a templated prompt. Radiology reports contain data and images that have low reader-interpretability, so GPT was employed to create more patient-friendly reports. The results were then manually analyzed and rated by experienced radiologists. This study found that both GPT-3.5 and GPT-4 achieved high performance in converting rawer data into easier to understand content, with the radiologists giving an average score of of 4.27 out of 5.

III. Methods

In this section, we present our methods in details. We use a dataset of real UAV images of waterfowl taken over the past year from different parts of Missouri to develop and test our methods [5]. Different landscapes are present in the images, like ponds, farmland, forest, grassland, etc. Image sequences were also taken at different times of day in various seasons of the year, causing for there to be different natural light levels and various landscape changes amongst the data. Each flight sequence contains between 4-20 images, and the sequences are sorted into four categories by altitude of flight relative to the ground: 15m, 30m, 60m, and 90m. It is important to note that consecutive images in 15m sequences don't contain significant overlap, while sequences from the other altitudes do. There are also many images in the dataset that do not contain any birds, and some sequences use decoy birds instead of real ones. In this paper, we use 5 image sequences captured at 30 meter height and 5 sequenced captured at 60 meter height to demonstrate our proposed methods and pipeline. Fig. 2 shows the first three images from three 30-meter sequences.

A. Image Overlap Detection

In order to avoid double-counting birds in overlapping regions between consecutive images we propose 3 methods to estimate the areas of the images that overlap.

1) GPS Based Method: The first method we propose is a geometric method that uses the GPS coordinates and other information about the drone camera in order to estimate the overlap. By extracting info from an image's metadata, we can obtain the GPS location of where each image was taken and therefore compute the distance between the images in the physical world. We can also use the metadata to calculate how much real world area each image is covering.

From a drone image's metadata, we obtain relevant information such as the focal length, make, and model of the camera. These are important for the Ground Sampling Distance (GSD) formula, which gives us the real world area covered by an aerial image. For example, if the GSD of an image is $5cm$, that indicates that each pixel represents a $5cm * 5cm = 25cm^2$ area in real life. Using GSD, we can compute the distance from the center of the image where the camera is positioned to the edge of the image. The GSD formula is

$$GSD = \frac{S \cdot H}{F \cdot w} \tag{1}$$

where

- S is the camera's sensor width;
- F is the camera's focal length;
- H is the height of the camera from the ground;
- w is the width of the taken image in pixels.

An illustration of the variables affecting GSD is shown in Fig. 3.

With GPS coordinates and the GSD, we know the distance between the centers of a pair of consecutive images and the

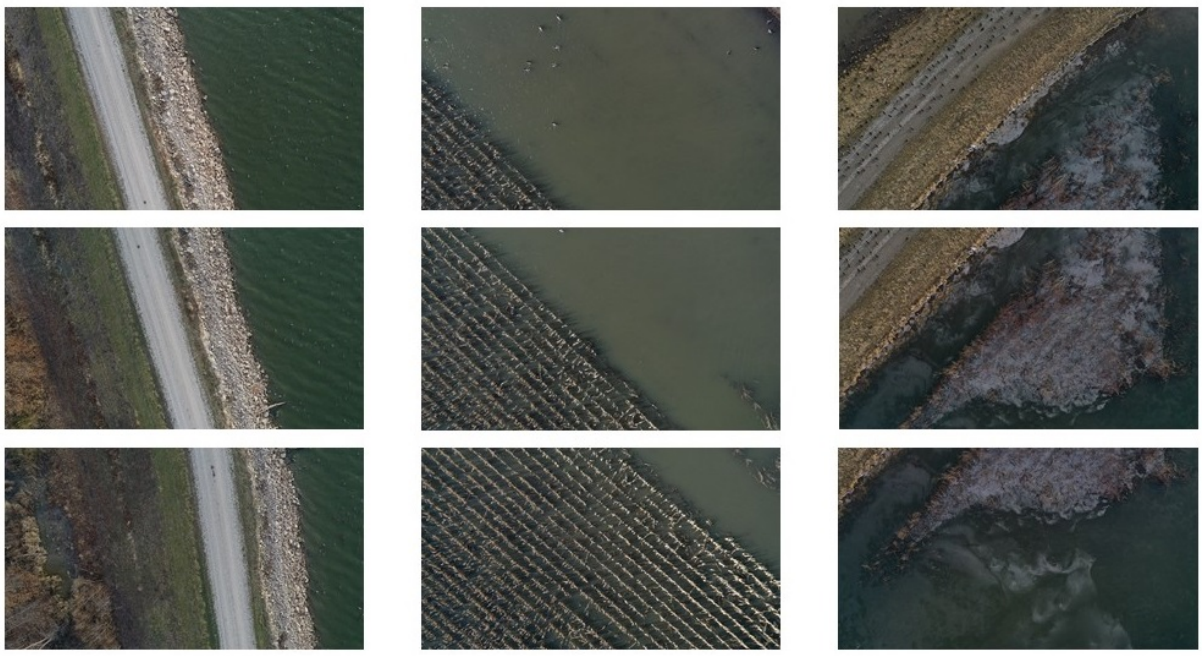

Fig. 2: The first three images from 3 different 30-meter image sequences in our dataset.

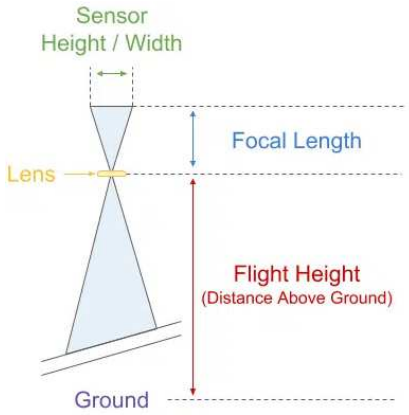

Fig. 3: An illustration of the variables affecting Ground Sampling Distance (GSD).

distances from the center of each image to its edge. To get the real-world distance between two images from GPS coordinates we use the following formula:

$$c = \sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos\left(\phi_1\right)\cos\left(\phi_2\right)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right) \quad (2)$$

$$D = 2000R\arcsin\sqrt{c} \quad (3)$$

where

- D is the distance between the two images in meters;
- ϕ_1 is the latitude in radians of the first image;
- ϕ_2 is the latitude in radians of the second image;
- λ_1 is the longitude in radians of the first image;
- λ_2 is the longitude in radians of the second image;
- R is the radius of Earth in Km (6371).

Using this conversion formula, we are able to get the direct distance between the two images, as well as the distances in the latitude and longitude components. To get the distance from the center of the image to one of its edges, we can do

$$D_x = GSD \cdot w/200 \quad (4)$$

$$D_y = GSD \cdot h/200 \quad (5)$$

where

- D_x is the distance from the center of an image to the left or right edge;
- D_y is the distance from the center of an image to the top or bottom edge;
- w is the width of the image in pixels;
- h is the height of the image in pixels;
- GSD is the Ground Sampling Distance.

Assuming that the drone stays at the same altitude throughout the flight, D_x and D_y should remain constant for all images within a sequence.

To get the overlapping area, we can simply subtract the distance from the center of one image to the edge of the same image from the distance from the center of one image to the center of the other image. This can be represented by

$$x = \frac{100(D_{lo} - D_x)}{GSD} \quad (6)$$

$$y = \frac{100(D_{la} - D_y)}{GSD} \quad (7)$$

where

- x is the offset of the overlap region in the horizontal axis in pixels;
- y is the offset of the overlap region in the vertical axis in pixels;
- D_{lo} is the distance between two images in the longitude component;
- D_{la} is the distance between two images in the latitude component;
- D_x, D_y, GSD are defined previously.

There are some limitations to this method. First, we need to determine the direction the camera is facing during the flight. Secondly, the formulas only give us the offsets of overlapping regions, not where they actually are on the image. We must know the direction of the drone flight relative to the direction of the camera, and then we can draw the bounding box on the image based on the obtained offsets that denotes the overlapping area. The corner we start the offsets depends on the direction of the flight. For example, if the direction of the flight is "rightup" and we had an x and y offsets of 300 and 500 respectively, we would draw a vertical line 300 pixels to the right of the bottom left corner and horizontal line 500 pixels above the bottom left corner to form the bounding box. Similarly, if the direction of the flight is "leftup", we would start from the bottom right corner.

2) SIFT Based Method: The second method we propose uses the well-known SIFT [12] algorithm to estimate the overlap region. For a more detailed description on how SIFT works, please refer to the referenced paper. In this method, we first transform the images to grayscale and use SIFT to detect features on the images. We then use KNN Feature Matching to match features between the images to compute the homography matrix from the matched points. We use the homography matrix to transform the points on the second image, and form a bounding box around the points using the furthest points on the top, right, bottom, and left. This bounding box is used to denote the overlapping area.

3) RANSAC Based Method: The final method we propose is similar to the SIFT method but uses RANSAC [13]. In this method RANSAC will randomly select points from anywhere on the image and we allow the algorithm to have a high limit of iterations. Once it reaches the limit, the best computed homography matrix is used. We then use SIFT to detect matching feature points like we did in the SIFT based method and apply the transformation and create the bounding box.

B. Waterfowl Detection and Counting

We run our existing bird detection model, a deep learning model trained using a large number of drone images on waterfowl [4], on each individual image in a sequence to find waterfowl. The model achieves waterfowl detection accuracy around 90% on drone images captured at 30 to 60 meter altitude. Once the overlap area between two consecutive images is determined, any birds detected within the estimated overlap regions are excluded to avoid double counting.

Fig. 4: An example of SAM segmentation result (bottom) on a drone image (top).

C. Habitat Segmentation and Classification

In addition to bird counting, we also classify the habitats within the image sequence and determine the distribution of birds across different habitats. First, we use Meta's latest segmentation model SAM [1] to segment each aerial image. SAM requires a prompt in the form of either points or bounding boxes. In order to give SAM an unbiased prompt, we place a grid of 64 evenly spaced points on an image as a prompt. SAM then automatically outputs pixel-by-pixel masks of what it recognizes as different regions of the image. Fig. 4 shows an example of SAM result.

Next, we trained machine learning models to classify the masks that SAM generates into 7 habitat categories. We modified the PyTorch's pre-built image classification model EfficientNet to perform classification on 224x224 pixel images from seven categories of natural region: *open water, herbaceous, shrub, cropland, harvested cropland, wooded, and other*. We retrained the model from the ImageNet pre-trained weight using supervised learning on a small manually labeled training set of around a thousand 224x224 pixel crops taken from the dataset. On the test set, the model achieved over 99% accuracy.

The segment masks generated by SAM can be of any shape or size. We divided each image into 224x224 pixel crops and classifying all of them, assigning each pixel a class label. To determine the class of each SAM segment, we simply use the majority class of the pixels in the segment.

D. ChatGPT Based Report Generation

The final stage of our new pipeline is to create an informative and comprehensive report about the drone flight result using ChatGPT. The report is based on the number of birds, their locations in the images, the classes of the various regions on images, which habitat each bird is in, the total area

covered by the flight using GSD, the percentage of the total area covered by each habitat class, and additional contextual information. Besides waterfowl and habitat information, we also extract data from the image metadata such as the date and start time of the flight, the total flight duration, the start and end GPS locations, and the flight altitude. Using the date, time, and GPS location, we are also able to gather data about the weather during the flight using OpenWeatherMap API.

Once we gather all these data, we use it to fill in a pre-built text template that is used to prompt OpenAI's GPT 3.5 Turbo model in order to generate the report, limiting the response to 2048 tokens.

IV. Experimental Results

In this section, we present some preliminary results. We use 5 image sequences captured at 30 meter height and 5 sequenced captured at 60 meter height to demonstrate our proposed methods and pipeline.

A. Overlap Detection

For evaluating our 3 proposed methods, we used Mean Absolute Error (MAE) to measure the degree of difference between Ground Truth (GT) and predicted overlapping area. The formula of MAE calculation are

$$AE_x = \frac{|x_{pred} - x_{gt}|}{W} \tag{8}$$

$$AE_y = \frac{|y_{pred} - y_{gt}|}{H} \tag{9}$$

$$MAE = \frac{AE_x + AE_y}{2} \tag{10}$$

where

- AE_x and AE_y are horizontal and vertical absolute error between prediction and GT;
- W and H are width and height of the image in pixels;
- x_{pred} and x_{gt} are coordinate x of right-up vertices of prediction and GT;
- y_{pred} and y_{gt} are coordinate y of right-up vertices of prediction and GT.

Tables I and II show the MAEs of the three overlap detection methods on each of the 5 image sequences captured at 30 meters and 60 meters, respectively. On the 30-meter test sequences, the RANSAC based method performed the best on average, being the best in 4 out of 5 cases. The GPS based method performed the best on Sequence #3. Their MAEs vary significantly across the 5 sequences. For example, the RANSAC based method achieved 0.1% error on Seqence #1, yet only 18.9% on Sequence #3. The performances of the RANSAC based method and SIFE based method are much better on 60-meter image sequences, achieving excellent results, less than 1% error. This is probably due to more overlaps between 60-meter images than 30-meter images. The performance of the GPS based method on 60-meter image sequences is slightly better, on average, than its performance on 30-meter sequences.

The execution time of the GPS based method is very fast, just a few milliseconds. The average execution times of the SIFT based method for the 30-meter and 60-meter sequences are 7.8 seconds and 19.1 seconds, respectively, on a Dell desktop computer. The average execution times of the RANSAC based method for 30-meter and 60-meter sequences are 41.4 seconds and 239.4 seconds, respectively. The RANSAC based method is much slower. It took longer for these two methods to process 60-meter images than to process 30-meter images, likely because 60-meter images have more content and are more complex.

Although the GPS based method outperforms the SIFT and RANSAC based method in terms of running speed, it is worse on prediction accuracy. The RANSAC based method has the best accuracy, but takes much longer to execute. The SIFT based method offers a good balance of accuracy and speed.

TABLE I: Performance comparison in terms of Mean Absolute Error of three overlap detection methods on five test 30-meter image sequences.

	Seq. 1	Seq. 2	Seq. 3	Seq. 4	Seq. 5
GPS-based method	7.1%	44.6%	10.4%	31.4%	3.4%
SIFT-based method	0.2%	88.0%	24.7%	20.6%	0.7%
RANSAC-based method	0.1%	3.4%	18.9%	15.2%	1.4%

TABLE II: Performance comparison in terms of Mean Absolute Error of three overlap detection methods on five test 60-meter image sequences.

	Seq. 1	Seq. 2	Seq. 3	Seq. 4	Seq. 5
GPS-based method	5.6%	18.1%	12.0%	8.3%	22.4%
SIFT-based method	0.04%	0.06%	0.15%	0.06%	0.10%
RANSAC-based method	0.04%	0.08%	0.10%	0.06%	0.10%

B. Habitat Classification

Our ground truths for habitat classification is based on a large number of test points on each image labelled with a habitat class. To evaluate the performance of our new SAM-based segmentation followed by image classification method, we compare the predicted classes of the test points with their labels and calculate prediction accuracy as the percentage of test points correctly predicted. It turned out that our method achieved almost perfect result on the test sequences, over 99% accuracy.

C. Report Generation

After generating and gathering all the necessary data about the flight path, waterfowl, and habitats, we use the data to fill in the blanks in pre-built text templates which are then used to prompt ChatGPT, i.e., GPT 3.5-Turbo. Fig. 5 show

rewrite the following report verbosely as a scientist, add any extra information about the location on the date you can find: Date: 19/12/2020 Temperatures were 22.73 degrees C. Precipitation was 22 mm. Humidity was 48%. Wind speed was 2.77 m/s. Weather description: light rain This flight's tally of 254 ducks. This flight covers a total area of 3285.10 sq. m. Out of this area, 55% open water, 20% cropland, 0% woody, 15% grassland, 5% road, 5% other. The ducks are distributed over the different habitats as follows; 80% on open water, 15% on cropland, 0% on woody, 0% on road, 5% on other. The flight was captured with a DJI Mavic Pro 2 drone equipped with a Hasselblad camera. It started at 39.7057333, -93.2798509, ended at 39.705147, -93.2799786 for a flight distance of 66.10m at an altitude of 29.7m.

rewrite the following report verbosely as a scientist, add any extra information about the location on the date you can find: data: date: 19/12/2020 start_time = 05:10:22 end_time = 05:10:34 start_gps = 39.7057333, -93.2798509 end_gps = 39.705147, -93.2799786 distance = 66.10 m altitude = 29.7 area = 3285.10 sq m water = .55 cropland = .20 woody = .00 grassland = 0.15 road = 0.05 other = 0.05 birds = 254 birds_on_water = .80 birds_on_cropland = .20 birds_on_woody = .00 birds_on_grassland = .15 birds_on_road = .00 birds_on_other = .05 weather: temperature = 22.73 C wind_speed = 2.77 m/s humidity = 48% description = "light rain"

Fig. 5: Two examples of our ChatGPT promote templates filled in with flight data.

examples of two different ChatGPT promote templates filled in with flight data. We have a basic criteria for what a good automatic report should accomplish:

- Present the data that was given in the prompt correctly;
- No content that is irrelevant to the report;
- Easy to understand, especially for conservationists;
- Reasonable length (1-2 pages);

From the experimental reports that have been generated, only 1 failed to meet this criteria, where the numbers regarding the bird count per habitat and the total area coverage by a class of habitat strangely got changed from the prompt to the report. An interesting note is that the content of the reports stays relatively consistent in different tries, while the structure of the report varies depending on how you ask ChatGPT to write the report. For example, asking ChatGPT to "write a report" gives you a standard text with separate paragraphs while asking it to "write a scientific repor" causes ChatGPT to respond with a report that includes an abstract and is divided into different headed sections.

V. Conclusions

In this paper, we present a new pipeline to process a sequence of drone images captured by a drone along a flight path to detect and count birds and their distributions in various habitats. We utilize customized deep learning models for waterfowl detection, Meta's SAM for image segmentation and customized image classification models for segment classification, and ChatGPT to generate text-based flight reports. Successful image overlap detection methods were developed to reduce waterfowl double counting in consecutive images.

In this work, we have uncovered how double-counting is a major problem in detecting objects in a sequence of aerial images. We have proved that detecting the overlap region between images in sequences and avoiding doing object detection in those regions is one viable solution to the double-counting problem. Out of our 3 proposed methods, we have found that our SIFT based method provides a good balance of accuracy and speed. Several factors could affect the performance of the overlap detection, such as having a lack of identifying features, small overlapping regions, a significant change of altitude during the flight, and other potential unidentified factors. Poor overlap detection will greatly distort the accuracy of the output. We have shown that a new, task-generalizing model in SAM performs well even on aerial images that it wasn't designed for. One limitation is that SAM requires a tremendous amount of GPU memory in order to run, especially with high resolution images. Finally, we found that ChatGPT performs well in automating the task of writing reports about drone flights.

Acknowledgment

The work is partially supported by NSF REU grant CNS-2243619 and a grant from Missouri Department of Conservation.

References

[1] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023.

[2] D. Chabot and C. M. Francis, "Computer-automated bird detection and counts in high-resolution aerial images: a review," *Journal of Field Ornithology*, vol. 87, no. 4, pp. 343–359, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/jofo.12171

[3] S.-J. Hong, Y. Han, S.-Y. Kim, A.-Y. Lee, and G. Kim, "Application of deep-learning methods to bird detection using unmanned aerial vehicle imagery," *Sensors*, vol. 19, no. 7, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/7/1651

[4] Y. Zhang, S. Wang, Z. Zhai, Y. Shang, R. Viegut, E. Webb, A. Raedeke, and J. Sartwell, "Development of new aerial image datasets and deep learning methods for waterfowl detection and classification," in *2022 IEEE 4th International Conference on Cognitive Machine Intelligence (CogMI)*, 2022, pp. 117–124.

[5] ——, "Development of new aerial image datasets and deep learning methods for waterfowl detection and classification," in *2022 IEEE 4th International Conference on Cognitive Machine Intelligence (CogMI)*, 2022, pp. 117–124.

[6] S. Bhatnagar, L. Gill, and B. Ghosh, "Drone image segmentation using machine and deep learning for mapping raised bog vegetation communities," *Remote Sensing*, vol. 12, no. 16, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/16/2602

[7] L. W. Tait, S. Orchard, and D. R. Schiel, "Missing the forest and the trees: Utility, limits and caveats for drone imaging of coastal marine ecosystems," *Remote Sensing*, vol. 13, no. 16, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/16/3136

[8] L. Tang, H. Xiao, and B. Li, "Can sam segment anything? when sam meets camouflaged object detection," 2023.

[9] Y. Wang, Y. Zhao, and L. Petzold, "An empirical study on the robustness of the segment anything model (sam)," 2023.

[10] T. Landeen, "Deep learning based image overlap detection," 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:67318140

[11] Q. Lyu, J. Tan, M. E. Zapadka, J. Ponnatapura, C. Niu, K. J. Myers, G. Wang, and C. T. Whitlow, "Translating radiology reports into plain language using chatgpt and gpt-4 with prompt learning: Promising results, limitations, and potential," 2023.

[12] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157 vol.2.

[13] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, jun 1981. [Online]. Available: https://doi.org/10.1145/358669.358692