Deep Learning for Overseeing Indo-Pacific Bottlenose Dolphin Tourism Law Enforcement

Dongwoo Kim[1,8*], Woohyun Jeon[2,8†], Jiyoung Lee[8†], Myeonghun Lim[3,8†], Hyemin Park[8†], Sunghun Yang[8†], Junhyung Kang[4,8†], Haneol Lee[5,8†], Jongsu Choi[8†], Dongki Chung[8†], Mi Yeon Kim[6‡], Soojin Jang[6‡], Seowoo Han[7,8§]

[1]Gwangju Institute of Science and Technology, [2]SELab Inc., [3]Superbin Co. Ltd, [4]SK C&C, [5]Korea University, [6]Marine Animal Research and Conservation, [7]VISIT INC., [8]DVA LAB, MODULABS

Email: {kimdw0611, jwh2204, ejiyoung96, roche5610, lemonbuilder, jk6385}@gmail.com, jh.kang@sk.com, roytravel@korea.ac.kr, jognsu@naver.com, dkc@kakao.com, {miyeonkim88, kesha00, swhan0329}@gmail.com

***Abstract*—In response to the threat facing Indo-Pacific bottlenose dolphins due to excessive tourism in Jeju island, South Korea, efforts were made by the Korean Ministry of Oceans and Fisheries to control boat distance and speed near dolphins. However, monitoring boat-dolphin distances via video proved challenging using the traditional methods. We propose an AI-driven solution using UAV-recorded videos to detect dolphins and boats, estimate distances, and monitor boat speeds. Our method, integrating a pipeline with YOLO v8, ByteTrack, and Efficient-AD, advances beyond prior single-frame ImageJ estimations by enabling continuous tracking and reducing false positives through video processing. Incorporating a Bird's-eye-view transformation, our pipeline enhances distance estimation between objects by providing an orthogonal virtual perspective. This approach not only aims to protect dolphins but also promotes responsible tourism practices, aligning with global efforts for wildlife conservation and sustainable tourism.**

***Index Terms*—Deep-learning, Tracking, Detection, Bird's-eye-view(BEV), Unmanned Aerial Vehicle(UAV), Distance estimation**

I. Introduction

Dolphins(*Tursiops aduncus*) in Jeju Island, South Korea, face severe threats due to excessive dolphin tourism. To combat this, the Korean Ministry of Oceans and Fisheries announced the Conservation and Management of Marine Ecosystems Acts that limit access to dolphins. These acts include regulations such as allowing no more than two boats to approach the dolphin groups within a 300m(meters) radius simultaneously and prohibiting all approaches within a 50m radius [1]. However, monitoring compliance proves challenging due to a lack of trained personnel to document the evidence of the violation and the absence of precise measurement of the distance between the dolphin and the boat as a shred of clear evidence to enforce the act. We propose a deep-learning method to detect dolphins and boats with Unmanned Aerial Vehicle(UAV)-recorded videos, accurately measuring the distance and speed of approaching boats. Compared to the distance estimation method using exclusively the sensor data where the error rate was over 40%, our pipeline which used AI models semi-automatically calculates the distance reducing the error rate within 5%. We expect our work to eliminate ambiguity in distance measurements during proximity encounters, thereby lending legal credibility to violation reports and ultimately contributing to the conservation of the dolphin population.

II. Related works

Deep learning techniques have been increasingly applied to wildlife conservation and monitoring. In the context of marine mammal protection, several studies have employed deep learning algorithms for detecting and tracking cetaceans in aerial imagery. Guirado et al. (2019) utilized a Faster R-CNN architecture to detect and count whales in satellite images, demonstrating the potential of deep learning for large-scale marine mammal surveys [2]. However, unlike the larger whales, which can be detected in static satellite images, our target species is relatively small and not easily recognizable using solely a detection model. Therefore, it requires the contextual information from consecutive frames to accurately predict its location.

Distance estimation between objects in aerial imagery has been a focus of several studies. Ribera et al. (2019) proposed a deep learning approach for estimating the distance between objects in a single image using a Siamese network architecture [3]. ImageJ, a popular open-source image processing program developed by the National Institutes of Health(NIH) and the Laboratory for Optical and Computational Instrumentation(LOCI) at the University of Wisconsin, can perform distance measurements on images. By setting a known scale using a reference object, users can determine distances or object sizes with its built-in measurement tools [4], [5].

However, these distance estimation approaches have limitations when dealing with large volumes of video data, as it requires manual annotation of individual frames and may not account for the movement of objects across multiple frames.

---

* is the first author.
† contributed equally to this work.
‡ provided with field data and consultation for Indo-Pacific bottlenose dolphins.
§ is a corresponding author and managed this project.

Moreover, since most videos are filmed from high altitudes up to 100m above the ocean surface, conventional methods like ImageJ face challenges in detecting objects and measuring distances accurately.

Bird's-eye-view transformations have been used to enhance object detection and distance estimation in aerial imagery. Zhu et al. (2021) proposed a method for transforming perspective views to bird's-eye views using a combination of semantic segmentation and depth estimation, improving object detection and localization in autonomous driving scenarios [6]. Nevertheless, the effectiveness of the method used in driving scenarios can be significantly reduced when applied in monitoring ocean surface, due to unfavorable conditions like light reflection and rapid changes in distance from the surface to the UAV.

Our method integrates advanced deep learning algorithms—YOLO v8, ByteTrack, and Efficient-AD—into a unified pipeline for detecting and tracking dolphins and boats in UAV-recorded videos. By incorporating a bird's-eye-view transformation, we enhance distance estimation accuracy, advancing AI-assisted wildlife monitoring and conservation.

III. Contributions

The main contributions of this paper are as follows:

1) We propose a novel deep learning-based method for detecting and tracking Indo-Pacific bottlenose dolphins and boats in UAV-recorded videos, incorporating state-of-the-art object detection (YOLO v8), multi-object tracking (ByteTrack), and anomaly detection (Efficient-AD) algorithms.
2) We introduce a bird's-eye-view transformation approach to enhance the accuracy of distance estimation between dolphins and boats, providing a more reliable means of monitoring compliance with marine conservation regulations.
3) Our method automates the process of analyzing large volumes of aerial video data, overcoming the limitations of manual annotation and single-frame analysis techniques, such as those based on ImageJ.

IV. Dataset

To address the absence of an open-source dataset for our task, we created our own using UAV recordings from Marine Animal Research and Conservation. Frequent use of the drone at a very close range ($\leq$10m over the animals) could significantly affect the behavior of the dolphins caused by noise and/or visual cues from the UAV or its shadow [7] [8] [9]. UAVs are currently limited to a minimum altitude of 30m over dolphins on Jeju Island unless a specific permit is obtained. Notably, all the videos used in the analysis were taken higher than 30m above the animals. Therefore, the impact of drone use on the dolphin population is anticipated to be minimal, especially when compared to the significant adverse effects of unregulated dolphin-watching vessels. The dataset consists of images, parsed from videos recorded from 2018 to 2023 with DJI Mavic 2 Pro. The video has a resolution of 3840×2160, in the RGB format. The frame rate of the videos is 30fps. Parsed images are saved in a single image folder and randomly split into a training set, a validation set, and a test set in a ratio of 6:2:2, respectively.

For the object detection task, the frame is parsed every 5 to 10 seconds, while it is parsed every 40 seconds for the anomaly detection task. The time interval of parsing for anomaly detection is set longer than that for object detection to contain more diverse environments in the dataset. For the tracking dataset, we divide the dataset into three cases: frames with boats only, frames with boats and dolphins, and frames with dolphins. Frames are parsed without time intervals so that we can assign the tracking ID to the moving objects associated with neighboring frames.

The dataset consists of approximately 1,000 frames for object detection, 200 frames for anomaly detection, and 700 frames for tracking tasks. By applying Slicing Aided Hyper Inference (SAHI) [10], frames are converted to smaller patches, resulting in an increase in the total number of patches compared to the original dataset size. Further details about the SAHI are explained in Section V-B.

Two classes, dolphins and ships, were manually labeled as bounding boxes for object detection and tracking tasks, and as segmentation masks for anomaly detection. Labeling policies were set for each class. To prevent models from confusing spray with the object, only dolphins below transparent ocean surfaces that are observable to the naked eye are labeled, while the ones which overlaid with spray are not. In cases where the border between the ship and the ocean is ambiguous due to light reflection, the bounding box area of the ship is minimized so that only the recognizable pixels are processed. Water spray is also excluded when labeling ships.

V. Methodology

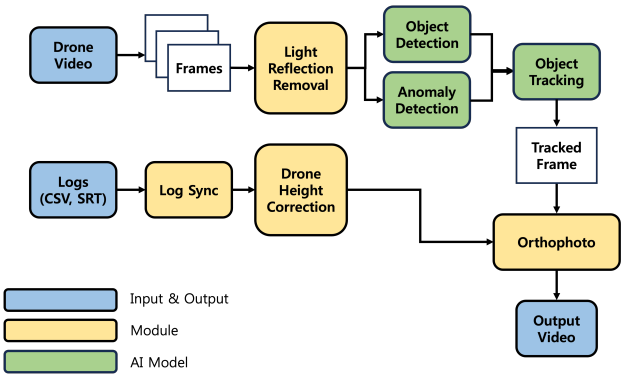

Fig. 1. Overall workflow of the pipeline

Figure 1 illustrates the overall workflow of our pipeline. The pipeline consists of mainly four parts:

1) Pre-processing the input data
2) The object detection and tracking
3) Conversion to Bird's-eye-view(BEV)
4) Distance & Velocity Estimation

Pre-processing involves parsing the input video into frames, applying light reflection removal, and synchronizing the video with drone log files. Frames are put into the object detection model and anomaly detection model to ensemble the results together. Merged inference results are then tracked by the tracking model. Next, the tracked result is converted into BEV for further operations of distance and velocity estimation. Finally, we visualize the output to ascertain whether any boats have violated regulations in specific frames.

A. Light Reflection Removal

Sunlight reflection on the ocean surface near Jeju Island often obscures dolphins from sight. To resolve this issue, we implement a queue frame fusion module, which inputs a certain amount of sequential frames in the queue, merging them as a hue, saturation, value(HSV) image as shown in Figure 2.

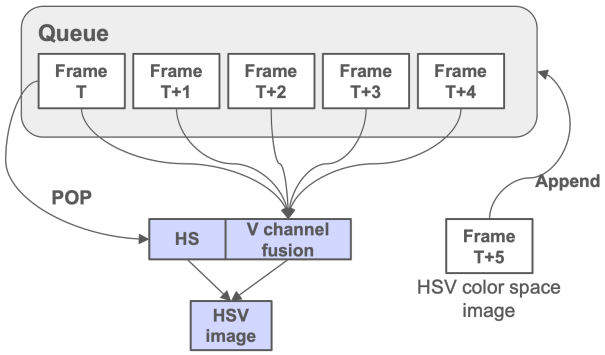

Fig. 2. The Workflow of Queue Frame Fusion

This approach effectively removes bright areas, while minimally affecting darker regions like the dolphins, thereby enhancing their visual distinctiveness. The application of these light reflection removal modules to the original footage, as demonstrated in Figure 3, results in improved detection performance by the AI model when compared to the unprocessed footage.

Fig. 3. The result of light reflection removal. (Left: Before applying algorithm, Right: After applying algorithm)

B. Object Detection

Because the video has a high resolution, we incorporated Slicing Aided Hyper Inference (SAHI) [10] during both training and inference to enhance the detection of small objects. Thus, it is significant to choose the most suitable slice size. By applying different slice sizes, the performance of YOLO v5 and YOLO v8 over our custom dataset is compared. The performance of different models and slice sizes are shown in Table I. Furthermore, by applying SAHI, we are able to increase the size of the dataset contributing to the performance of the models.

TABLE I
PERFORMANCE COMPARISON BETWEEN YOLO V5 AND V8

Image size	YOLO v5	YOLO v8
512	82.6%	74.8%
1024	83.3%	85.5%
1440	82%	79.3%
1920	77.2%	82%

Among the results, our tests revealed that a slice size of 1024×1024 with overlap ratio of 0.2, optimally detects objects in the UAV frames for both YOLO v5 and v8, mAP score of 83.3% and 85.5% respectively. Thus YOLO v8 is selected as a detection model for our pipeline, which scored the highest. The confidence threshold of the YOLO v8 is set to 0.3.

C. Anomaly Detection

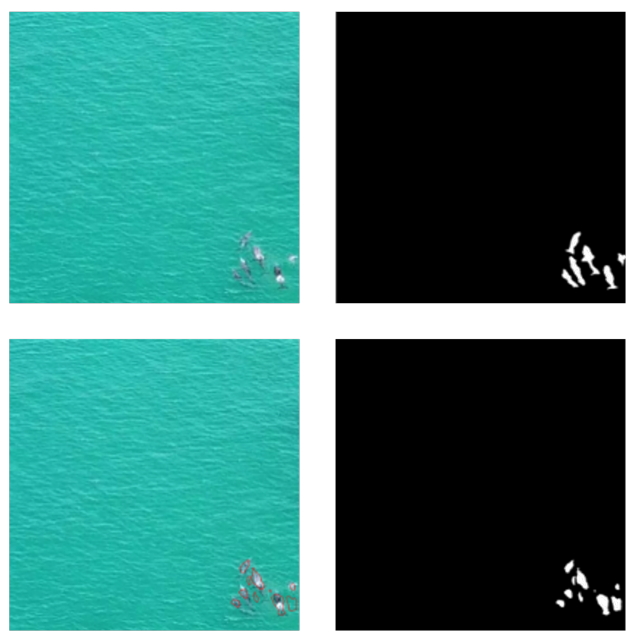

Fig. 4. Dataset and Inference of Anomaly Detection(Top Left: Input image, Top Right: Ground truth mask, Bottom Left: Segmentation result, Bottom Right: Predicted anomaly mask)

For preventing the detection of false-positive instances, Efficient-AD [11] is applied in the pipeline along with the object detection model. The model is trained to treat dolphins as anomalies while considering other objects such as boats, oceans, and reefs as normal. The anomaly masking threshold is set to 0.5, where pixels with an anomaly score above 0.5 are mapped as a value of 1, indicating an anomaly, and 0 otherwise.

Figure 4 depicts an example of anomaly detection. The input image (top left) shows a group of dolphins in the ocean. The ground truth mask (top right) highlights the dolphins. The segmentation result (bottom left) accurately identifies the dolphins. The predicted anomaly mask (bottom right) flags the dolphins as anomalies.

D. Merging Outputs

The outputs of the object detection and anomaly detection are then merged and saved in a single file. The ensemble process is necessary, as the performance of anomaly detection was not as robust as that of object detection. The outputs of YOLO v8 and Efficient-AD are grouped by frame as dictionaries, with the frame being the key and lists of detected boxes being the values. For each frame, the coordinate information of the bounding boxes is merged. Anomaly detection results are appended, but only those with confidence scores above the threshold are included. The relatively high threshold value is set so that anomaly results with high confidence scores are applied in the decision of bounding boxes. In our pipeline, we use a threshold value of 0.7. Algorithm 1 illustrates the overall process of the ensemble.

Algorithm 1 Ensemble Object Detection and Anomaly Detection

Require: anomalyDict: dictionary, where keys are frame numbers and values, are lists of anomaly detection boxes, each box is [frame number, x, y, w, h, probability]
Require: detectionDict: dictionary where keys are frame numbers and values are lists of object detection boxes, each box is [class, x, y, w, h, probability]
Require: useAnomaly: boolean flag indicating whether to use anomaly detection results

```
result = []
for frameNum in detectionDict.keys() do
    for detBox in detectionDict[frameNum] do
        result.append(detBox)
    end for
    if useAnomaly and frameNum in anomalyDict then
        for anomBox in anomalyDict[frameNum] do
            if anomBox[5] ≥ threshold then
                result.append(anomBox)
            end if
        end for
    end if
end for
```

E. Tracking

The merged results from YOLO v8 and Efficient-AD are then fed into the ByteTrack [12] model for object tracking. The tracking confidence threshold is set to 0.2, the track buffer to 30, and the matching threshold for tracking to 0.8.

F. Bird's-Eye-View Conversion

Converting frames to BEV [13] is vital for accurately measuring the distance between the dolphins and the boat.

Fig. 5. The result of BEV Conversion.(Left: Before applying BEV, Right: After applying BEV)

By transforming pixels to Ground Positioning System (GPS) coordinates and obscuring the virtual orthogonal view as shown in Figure 5, we can precisely calculate the real size of the pixels. In the pipeline, we let the user select the initial UAV frame for BEV to be applied, where reference scale like the length of the boat can be optionally added to gain more accuracy in the distance estimation process, which is further discussed in Section V-H.

G. Drone Height Correction

To accurately calculate the Ground Sample Distance (GSD), the height from the ground level to the drone is adjusted. Equation (1) shows how the adjusted height h_{adj} is computed, by summing up from (1a) to (1c), and subtracting (1d).

$$
\begin{aligned}
h_{\text{adj}} = h_{\text{rel}} & \quad \text{(1a)} \\
+ (L_{\text{max}} - L_{\text{obs}}) & \quad \text{(1b)} \\
+ (h_{\text{home}} - h_{\text{khoa}}) & \quad \text{(1c)} \\
- (H_{\text{drone}} - H_{\text{khoa}}) & \quad \text{(1d)}
\end{aligned}
$$

h_{rel} is the drone's relative height compared to the home point, obtained from the GPS information (1a).

L_{max} is the maximum tide level observed at the tide observation station, while L_{obs} is the current tide level observed at the tide observation station.

h_{home} is the absolute height of the home point, and h_{khoa} is the absolute height of the nearby Korea Hydrographic and Oceanographic Agency (KHOA) station.

H_{drone} is the height of the geoid (mean sea level) at the drone's observation area, obtained from the National Geographic Information Institute (NGII). H_{khoa} is the height of the geoid at the KHOA station.

Equation (1b) represents the difference between the maximum and current tide levels to account for the tidal variation. The difference between the absolute heights of the home point and the KHOA station is denoted as (1c) to reference the drone's height to the KHOA station.

The difference between the geoid heights at the drone's observation area and the KHOA station is denoted as (1d) to account for the variation in mean sea level. Assuming that the height of the sea is identical, we ignore this value.

H. Distance Estimation

The distance between the ship and the herds of dolphins is deduced through the application of the GSD metric which is

the product of the pixel size and the vertical distance from the camera to the sea surface, normalized by the camera's focal length. In the determination of pixel size, two distinct methodologies are compared. The first method involves dividing the physical width of the camera sensor by the number of pixels in the image width. On the other hand, the second method utilizes a known measurement such as the length of the car as a reference scale.

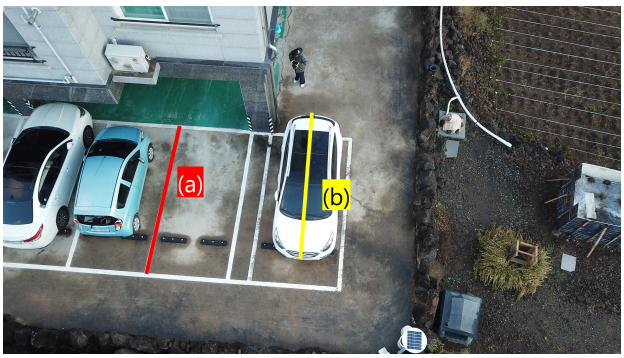

Fig. 6. Designated benchmark measurement (a) and a reference scale (b)

Fig. 7. Examples of selected frames with a measurement target

For an empirical evaluation of these methods regarding the accuracy of distance estimation, we set the ground truth value 4.85m for a certain measurement, the height of the rectangular parking area delineated by the white line. The space is captured with a UAV, filming the area from various angles and altitudes. Designated benchmark measurement is annotated as (a) in Figure 6. In the experiment, the length of the Hyundai Tucson car is used as a reference scale which is approximately 4.64m. Among the successive UAV frames, 20 frames that include the target area are chosen for estimation. The frames are shown in Figure 7.

TABLE II
COMPARISON OF DISTANCE ESTIMATION METHODS

	GT	1	2	3	4	5	6	Avg Loss %
M1	4.85	8.73	3.21	7.87	4.70	2.09	5.40	41.20%
M2	4.85	4.21	4.80	4.55	4.71	4.63	4.95	4.98%

The distance estimation results are presented in Table II, where 6 of the 20 frames' measurements are selected and displayed in meters. The 6 measurements' average loss is also computed and displayed as a percentage. "Method 1(M1)" represents the method that utilizes camera parameters for calculations, and "Method 2(M2)" refers to the method that employs a known reference scale. It is evident from the comparison that the use of a reference scale in method M2 results in a significantly lower average loss percentage of 5% compared to 40% for method M1. This underscores the enhanced accuracy when employing a reference scale in the GSD calculation pipeline.

The distance from the ship to the herd of dolphins is determined by measuring the center of the pod and the ship's bounding boxes. This is since sometimes multiple tourist ships approach the dolphins, and depending on the position of the individual ship, the dolphin located in the center of the distance calculation would change. By using the center of the pod, we ensure that the measurements are consistent for all cases, regardless of the position of the ships.

I. Velocity Estimation

$$V(km/h) = \frac{dist((x_1, y_1), (x_2, y_2))}{\Delta t} \times 3.6 \tag{2}$$

With the use of drone metadata, (e.g. geographic coordinates of drone, tilt angle) and GSD value, it is possible to derive geographic coordinates of every pixel in each frame. Then, the speed of the ship can be estimated by measuring the spatial distance derived from the change in geographic coordinates of the center of the ships over successive frames.

Due to Earth's curved surface, Equation 2 calculates the haversine distance between the two center points using their latitude and longitude coordinates, divides it by the time interval to obtain the speed in meters per second, and then multiplies it by 3.6 to convert the speed to kilometers per hour (km/h). (x_1, y_1), (x_2, y_2) represent the latitude and longitude coordinates of the bounding box's center point of the first and second frame, respectively. *dist* represents the haversine distance calculation, and Δt represents the time interval between the two frames. The result is rounded to 2 decimal places.

VI. RESULTS

To our knowledge, no existing systems directly compare to our work. However, compared to previous UAV-based marine mammal detection methods, like [14], our approach offers significant advantages. While past studies focus solely on detecting marine species, our system not only matches their detection performance but also distinguishes between dolphins and boats, enhancing its utility for complex marine surveillance and monitoring tasks.

A. Quantitative Results

As shown in Table III, our detection model scored a mean Average Precision(mAP) value of 85.5%. Efficient-AD attained 0.83 in the pixel-level area under ROC(AUROC). ByteTrack achieved its best Average Precision(AP) score at 84.94% for the evaluation.

TABLE III
PERFORMANCE OF AI MODELS

Detection (YOLO v8)	Tracker (ByteTrack)	Anomaly Detection (Efficient-AD)
mAP: 85.5%	AP: 84.94%	pixel_AUROC: 0.83

B. Visualized Output

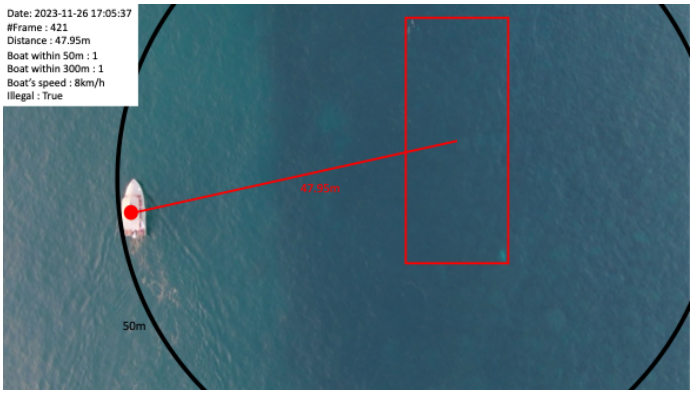

Fig. 8. Visualized output of the pipeline. Herd of dolphins are grouped as a single red bounding box

Figure 8 presents the visualized results with merged bounding boxes and estimated distance. The distance is noted along with the circle radius of 50m, stipulated in the regulation.

VII. CONCLUSION

The model demonstrates proficiency in detecting dolphins and boats from videos. Therefore, we submitted a visualized report of our findings to the Jeju Special Self-Governing Government Provincial Office, receiving positive feedback. The current detection and tracking limitations likely stem from the small dataset, composed of about 2,000 images. Given the challenges of obtaining data from violating scenarios, our dataset is not sufficiently robust for generalization. In response, we plan to enhance the variety of our dataset through the application of data augmentation in subsequent experiments. Increasing the dataset's diversity and size is expected to improve the model's precision significantly. We anticipate the project to emphasize the critical role of AI in addressing pressing environmental challenges and advancing global sustainability goals.

VIII. FUTURE WORK

Future developments include deploying this technology as a web service to provide conservation groups with quantitative trustworthy data. To efficiently use this pipeline in the service, we are planning to lightweight the deep-learning model. This expansion has the potential to minimize human disturbances significantly, promoting sustainable tourism practices. Such advancements not only contribute to the conservation management of Indo-Pacific bottlenose dolphins in Jeju island but also exemplify the use of AI in enforcing environmental regulations and fostering community engagement in conservation efforts.

The application of the AI system to different locations, species, and environmental conditions requires further research. We are currently investigating the system's potential to detect various dolphin species commonly found in Korean waters, such as finless porpoises, common dolphins, and Pacific white-sided dolphins.

ACKNOWLEDGMENT

This research was supported by Brian Impact Foundation, a non-profit organization dedicated to the advancement of science and technology for all.

REFERENCES

[1] Ministry of Oceans and Fisheries. Conservation and management of marine ecosystems act no 637, 2023.
[2] E. Guirado, S. Tabik, M.L. Rivas, D. Alcaraz-Segura, and F. Herrera. Whale counting in satellite and aerial images with deep learning. *Scientific Reports*, 9(1):1–12, 2019.
[3] J. Ribera, D. Guera, Y. Chen, and E.J. Delp. Estimating the distance between objects in a single image using a siamese network. In *27th European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE, 2019.
[4] C.A. Schneider, W.S. Rasband, and K.W. Eliceiri. Nih image to imagej: 25 years of image analysis. *Nature Methods*, 9(7):671–675, 2012.
[5] T. Ferreira and W. Rasband. Imagej user guide. *ImageJ/Fiji*, 1:155–161, 2012.
[6] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei. Bird's-eye-view panoptic segmentation using monocular frontal view images. *arXiv preprint arXiv:2108.03227*, 2021.
[7] J. Castro, F. O. Borges, A. Cid, M. I. Laborde, R. Rosa, and H. C. Pearson. Assessing the behavioural responses of small cetaceans to unmanned aerial vehicles. *Remote Sensing*, 13(1):156, 2021.
[8] T. Fettermann, L. Fiori, M. Bader, A. Doshi, D. Breen, K. A. Stockin, and B. Bollard. Behaviour reactions of bottlenose dolphins (tursiops truncatus) to multirotor unmanned aerial vehicles (uavs). *Scientific Reports*, 9(1):8558, 2019.
[9] C. E. Smith, S. T. Sykora-Bodie, B. Bloodworth, S. M. Pack, T. R. Spradlin, and N. R. LeBoeuf. Assessment of known impacts of unmanned aerial systems (uas) on marine mammals: data gaps and recommendations for researchers in the united states. *Journal of Unmanned Vehicle Systems*, 4(1):31–44, 2016.
[10] Fatih Cagatay Akyon, Sinan Onur Altinuc, and Alptekin Temizel. Slicing Aided Hyper Inference and Fine-tuning for Small Object Detection. *2022 IEEE International Conference on Image Processing (ICIP)*, pages 966–970, 2022.
[11] Kilian Batzner, Lars Heckler, and Rebecca König. Efficientad: Accurate visual anomaly detection at millisecond-level latencies, 2023.
[12] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box, 2022.
[13] Lin-Bo Luo, In-Sung Koh, Kyeong-Yuk Min, Jun Wang, and Jong-Wha Chong. Low-cost implementation of bird's-eye view system for camera-on-vehicle. In *2010 Digest of Technical Papers International Conference on Consumer Electronics (ICCE)*, pages 311–312, 2010.
[14] Eyal Bigala, Opher Bar Natan, Asaf Levy, Ori Galil, Massimiliano Rosso, Dan Tchernov, Tali Treibitz, and Aviad P. Scheinin. Automated detection of dolphins in imagery from unmanned aerial vehicles and performance optimisation. 2022.