Title

RipScout: Realtime ML-Assisted Rip Current Detection and Automated Data Collection using UAVs

Journal

IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 18

Authors

Khan, Fahim Hasan
Stewart, Donald
de Silva, Akila
et al.

Publication Date

2025-02-18

DOI

10.1109/JSTARS.2025.3543695

Peer reviewed

RipScout: Realtime ML-Assisted Rip Current Detection and Automated Data Collection Using UAVs

Fahim Hasan Khan, Donald Stewart, Akila de Silva, Ashleigh Palinkas, Gregory Dusek, James Davis, *Senior Member, IEEE*, and Alex Pang

***Abstract*—This article presents RipScout, a system for realtime rip current detection and data collection using drones equipped with machine learning (ML). No internet connection is required. RipScout achieves realtime performance by using lightweight ML models that fit the constraints of limited mobile computing resources with the drone controller. We compared different ML models trained to detect either one or two types of rip currents. The best model was then selected for RipScout. The system was evaluated with three ML models, with the EfficientDet D2 model achieving the highest accuracy of 93.1% for multiclass detection while maintaining realtime processing at an average speed of 17 frames per second. When a rip is detected along a flight path, the drone hovers in place and collects a video clip of a predefined length, followed by circling around the detected rip using prespecified radii and heights to collect video samples from different vantage points and elevations. An important benefit of RipScout is that the collection of rip current data can be performed by drone operators who are not familiar with rip currents. We conducted field tests and found that the proposed system allows data to be collected four times faster than without it while improving accuracy. As a by-product of the field experiments, we also provide a new rip current dataset. Such a multiviewpoint dataset can be used to improve rip current detection, especially from lower elevations and different orientations.**

***Index Terms*—Computer vision, data collection, machine learning (ML), mobile computing, real-time processing, rip current detection, uncrewed aerial vehicle (UAVs).**

Received 12 May 2024; revised 28 October 2024 and 23 December 2024; accepted 3 February 2025. Date of publication 18 February 2025; date of current version 21 March 2025. This work was supported in part by the Southeast Coastal Ocean Observing Regional Association (SECOORA), National Oceanic and Atmospheric Administration (NOAA) under Grant NA20NOS0120220, in part by the California Sea Grant and US Coastal Research Program (USCRP) as administered by the US Army Corps of Engineers(USACE), Department of Defense under Grant NA23OAR4170121, and in part by University of California, Santa Cruz Faculty Research Grant. *(Corresponding author: Fahim Hasan Khan.)*

Fahim Hasan Khan is with the Computer Science and Software Engineering Department, California Polytechnic State University, San Luis Obispo, CA 93407 USA (e-mail: fkhan19@calpoly.edu).

Donald Stewart, James Davis, and Alex Pang are with the Computer Science and Engineering Department, University of California, Santa Cruz 95064 USA (e-mail: dolstewa@ucsc.edu; davis@cs.ucsc.edu; pang@soe.ucsc.edu).

Akila de Silva is with the Computer Science Department, San Francisco State University, San Francisco, CA 94132 USA (e-mail: desilva@sfsu.edu).

Ashleigh Palinkas is with the California Sea Grant, Scripps Institution of Oceanography, University of California, San Diego, CA 92093 USA (e-mail: apalinkas@ucsd.edu).

Gregory Dusek is with the NOAA National Ocean Service, Silver Spring, MD 20910 USA (e-mail: gregory.dusek@noaa.gov).

Digital Object Identifier 10.1109/JSTARS.2025.3543695

I. Introduction

THIS article introduces RipScout, a machine learning (ML)-assisted system designed for efficient field data collection of rip currents using drones. Rip currents are dangerous, strong, fast-moving currents that pull swimmers away from the shore, often leading to drownings and fatalities. They pose a significant hazard to beachgoers and can easily overpower even strong, experienced swimmers. Rip currents are a global issue, affecting coastlines around the world [1], [2], [3]. Rip currents are responsible for a significant portion of beach-related drownings and rescues worldwide. In the United States alone, rip currents account for an estimated 100 fatalities yearly. The United States Lifesaving Association reports that 81.9 % of the 37 000 beach rescues each year are due to rip currents [4]. For example, in 2019, lifeguards in California reported over 3000 rescues due to rip currents, underscoring the persistent danger they pose to beachgoers. The impact extends internationally, with Australia recording an average of 21 rip-related drownings annually and thousands of near-drowning incidents. Case studies from the National Oceanic and Atmospheric Administration (NOAA) show that rip currents are incredibly challenging for untrained swimmers, who may not recognize the subtle signs of these currents until they are caught in one [5]. Rip currents can form suddenly and without obvious signs, which can catch swimmers off guard. While there are general conditions that can lead to their formation, predicting exactly when and where they will appear is challenging. Furthermore, rip currents are created through various mechanisms and, as a result, exhibit different visual characteristics. This complexity of occurrence and variability in appearance make them difficult to identify [6]. Consequently, many beachgoers lack the essential knowledge and awareness needed to recognize and avoid these perilous currents.

It is important to monitor beaches for the presence of these rip currents to improve beach safety. Data collected about rip currents are also important for validation studies of rip forecast models [7]. While traditional rip current data collection involves in-situ instrumentation, such as GPS-equipped drifters and current meters [8], [9], recent work has demonstrated that rip currents can be detected from images and video [10], [11], [12], [13], [14], [15], [16], [17]. Since cameras are more widely available than specialized instruments, these methods have the potential to greatly expand the amount of rip current data collected.

Uncrewed aerial vehicles (UAVs) such as drones have been widely used to perform visual data gathering tasks for environmental monitoring, search and rescue, and target tracking because they allow free-form flight maneuverability and thus versatile camera placement [18], [19]. Cameras on drones also have a distinct advantage of spotting rips from a higher elevation since some rips are difficult to see from a lower elevation. This article investigates whether a system can be created to detect and collect data on rip currents using drones that perform as effectively as domain experts, such as lifeguards, or leading rip detection algorithms. Formally, the research question for this work is: *Can a UAV-based system, equipped with lightweight ML models, reliably detect rip currents in real-time under the computational and resource constraints typical of mobile platforms?*

We first identify the technical challenges imposed by a mobile drone platform. These challenges include the ability to *detect rip currents in realtime* in order to support *field deployment*. The detection system must be *lightweight and run on limited computational resources* available on mobile platforms. Beaches are often remote with *no internet connectivity*, so sending images to a server for processing may not be possible. Given that battery capacity of consumer grade drones is typically in the 15–25 min range per charge, data collection needs to be *efficient*. And to be of value, the rip detection must have *sufficient accuracy*.

We then designed a system that seeks to address these technical challenges. Our proposed solution is that a real-time, ML-assisted UAV system can achieve high accuracy in detecting rip currents while operating within limited computational and battery resources. In our design, the mounted camera on the drone streams image data to a mobile phone. The mobile phone performs rip current detection and sends flight control signals to the drone. When a rip current is detected, the drone is directed to take the desired data collection activity, such as recording 30 s of video from several different altitudes and/or video recording of the rip from different angles and elevations. Our proposed system architecture is practical and portable, leveraging the increasing availability of high-quality drones [20] and the ubiquity of smartphones. The system is described in “System Architecture” and “Datasets” sections.

The system is then field tested and evaluated on how well it met its goal of accurate rip current detection and efficient data collection. To do so, we first narrowed down the choice for a base ML detection model (see “Related Works” section). Those that required powerful graphics processing units (GPUs) that are not available on mobile platforms were either removed from further consideration or modified to run on central processing units (CPUs). There were three ML detection models that were evaluated and the best performing one selected for RipScout. Our tests show that the proposed system can perform as well as a domain expert in terms of accuracy and efficiency. As designed, the system is also agnostic to specific ML models, as long as it can run within the resource constraints. Hence, the ML model employed for rip detection can be replaced with a better one in the future. The field tests, selection, and evaluations are described in the “Field Testing” and “Evaluation” sections.

- The primary contribution of this article is a drone-based system architecture called RipScout that allows efficient rip current data detection and data collection in a resource-constrained environment.
- The rip current data that were collected during field experiments are valuable for further rip current research.

II. Related Works

Lightweight ML Models for Drones

We review recent literature to look for candidate ML models that are potential candidates for use in rip current detection. A series of VisDrone challenges [21], [22], [23], [24] have focused on improving the accuracy and speed of object detection and tracking. The deep learning models in those papers can be divided into one-stage and two-stage structures. Recent trends in the VisDrone-DET 2019-21 challenges [21], [22], [23] show that one-stage models prioritize prediction speed in frames per second (FPS), while two-stage models strive for the best per-frame accuracy but incur higher computational costs.

The two-stage region-based detectors, such as faster region convolutional neural network (R-CNN) [25], typically include a region-proposal stage followed by a classifier. However, one major drawback of these detectors is their computational complexity, which makes them challenging to deploy on embedded platforms. For instance, faster R-CNN has been successfully used for realtime object detection with drones by connecting to a remote GPU server [26]. However, this approach is not practical in locations without internet connectivity.

In contrast, one-stage techniques employ a single CNN to perform end-to-end object detection [27]. This single CNN architecture render the models lightweight and able to run quickly on mobile devices, making it an ideal choice for realtime object detection applications. The main tradeoff for this speed is a decrease in accuracy compared to more complex models [28]. Sun et al. [29] demonstrated that single CNN architectures designed for mobile devices also require the least amount of memory compared to other CNNs.

As such, we focus our review on the latest one-stage ML models to determine suitable candidates for our system. We found SSD-MobileNetV2 [30] as one of the first reliable models that work well for mobile systems. However, the top choices from the most recent one-stage models optimized for mobile devices are EfficientDet [31], and you only look once (YOLO) [32]. Detection Transformers [33] was also briefly considered but was found to have a significantly slower inference speed and require a server with GPU. A comparative study by Mekhalfi et al. [34] demonstrated that EfficientDet is the more stable and offers good generalization capability for applications with aerial imagery. Even though the newest YOLOv8 (2023) improved over its predecessor in accuracy, EfficientDet still provides more stability while achieving similar or higher accuracy [35].

Field Tested Drone Applications with ML

Field testing is needed to ensure that results from experiments in controlled lab settings transfer and actually perform in real-world scenarios. In many drone applications, it is crucial to have a model that can process and predict data quickly, approaching the speed at which the drone transmits video. For example, the drone used in this work transmits images at 30 FPS [36]. However, few research papers address the challenge of embedding models in real-world settings [37], [38]. For example, while frame accuracy of an ML model may appear very good when compared against known ground truths, deployment in real-world where other factors such as fog, haze, glare, vibrations, etc., come into play may be quite different.

Although some studies [39], [40] do present empirical studies on embedded UAVs to perform realtime visual object detection and tracking, they have not been validated through field tests, which are crucial for evaluating the performance of such systems in real-world environments.

In recent times, several projects have utilized ML models to process drone images. However, these projects vary in terms of where the processing takes place. For instance, Rohan et al. [41] presented a drone-based system that employed a lightweight SSD model on a desktop system equipped with a high-end GPU. Nevertheless, their system is not feasible for field deployment, and they only conducted tests indoors. On the other hand, Deng et al. [42] deployed a similar lightweight model on the Nvidia Jetson TX2 Board, a specialized embedded platform that can be mounted on a drone. While this platform offers suitable performance, it is not as widely accessible as mobile phones, thus limiting its user base. Similarly, another project utilized the same GPU board for a UAV-based warning system with realtime object detection [43]. However, both of these projects did not conduct real-world field deployment and instead tested their system using existing drone footage.

Vajgl et al. [44] described a pattern-matching algorithm for drone control that runs on a mobile device and tested it to detect simple objects in a lab setting. None of these projects attempted to perform complex tasks such as realtime object detection for data collection in real-world settings. The differences in processing locations and the limitations in real-world deployment highlight the need for a more versatile and accessible approach to drone image processing and object detection.

While numerous studies have demonstrated the potential of UAVs in environmental monitoring, search and rescue, and target tracking, several limitations remain. Many prior UAV applications lack real-time processing capabilities or require a remote server for data analysis, which is impractical in remote locations without internet connectivity. High power demands often limit other UAV-based approaches and need more efficient data collection strategies, resulting in restricted operational times due to limited battery life. Furthermore, previous studies inadequately address the complexities of detecting subtle, dynamic natural phenomena, such as rip currents, which present unique challenges due to their varied appearance and transient nature. In contrast, we introduce RipScout, a real-time, ML-driven system designed to address these gaps. By integrating lightweight ML models capable of processing on mobile devices, RipScout eliminates the need for server dependency, achieving faster rip current detection and automated data collection directly on the drone's platform. This design enables nonexpert operators to collect data, enhancing accessibility and efficiency in rip current monitoring. In this article, we evaluated our proposed system with several field tests described in the "Field Testing" section.

A. *Rip Current Detection With ML*

There has been a growing interest in remote detection of rip currents using images and/or video in recent years. We review and organize the literature by how suitable the approach might be for deployment on a mobile platform. In particular, we consider if images/videos from a stable platform are required and whether the ML model requires significant computing power, i.e., GPU.

The problem of rip current detection has been addressed using various methods, including approaches predating the emergence of deep learning techniques. Philip et al. [15] employed optical flow on video sequences to identify the predominant flow toward the sea, assisting human observers in rip current detection. Maryan et al. [13] utilized modified Haar cascade methods to detect rip currents from time-averaged images. Silva et al. [10] were among the early adopters of deep learning methods for rip current detection, employing faster R-CNN, a two-stage model that achieved high accuracy. They also proposed a frame aggregation technique that improved detection accuracy for fixed-position cameras, which is not applicable to moving cameras, such as those mounted on drones. [14] proposed a non-ML, flow-based method for highlighting and visualizing rip currents by showing the behavior of a set of virtual buoys (pathlines and timelines). Similarly, RipViz [11] analyzes 2-D vector fields and uses ML to learn pathline behavior and automatically identify rip currents. This method highlights the *shape* of the rip region. Detection is based on water movement behavior rather than on the appearance of the water state. All the above-mentioned methods require video from a stable platform and hence are not directly applicable for analyzing drone videos.

Rashid et al. [17] and Zhu et al. [45] introduced RipDet and YOLO-Rip, lightweight rip current detection models based on Tiny-YOLOv3 and YOLOv5s, respectively. These are smaller models in the YOLO family and well-suited for running with limited computational resources. These models have since been superseded by YOLOv8. In fact, [12] utilized and compared various versions of YOLOv8 for rip current segmentation. While YOLOv8 can theoretically run in realtime on both desktop and mobile system, Dumitriu et al. [12] did not discuss an implementation on a mobile system. Rampal et al. [16] demonstrated that the mobile-optimized single-stage model SSD-MobileNetV2 can achieve comparable accuracy to faster R-CNN. However, they used an Nvidia P100 GPU which has a maximum power draw of 250 W and has a large form factor and therefore cannot be used in any smartphone. Further investigation revealed that it is possible to implement MobileNetV2 to run on a CPU with real-time results.

Apart from rip current detection, ML has also been used for applications, such as predicting rip current vulnerability. A study by Najafzadeh et al. [46] leveraged nine ML models to predict rip current vulnerability on southern Chinese beaches, identifying high-risk areas for strategic management and demonstrating the superior accuracy of the multivariate adaptive regression spline (MARS) model in estimating relative tide range (RTR)

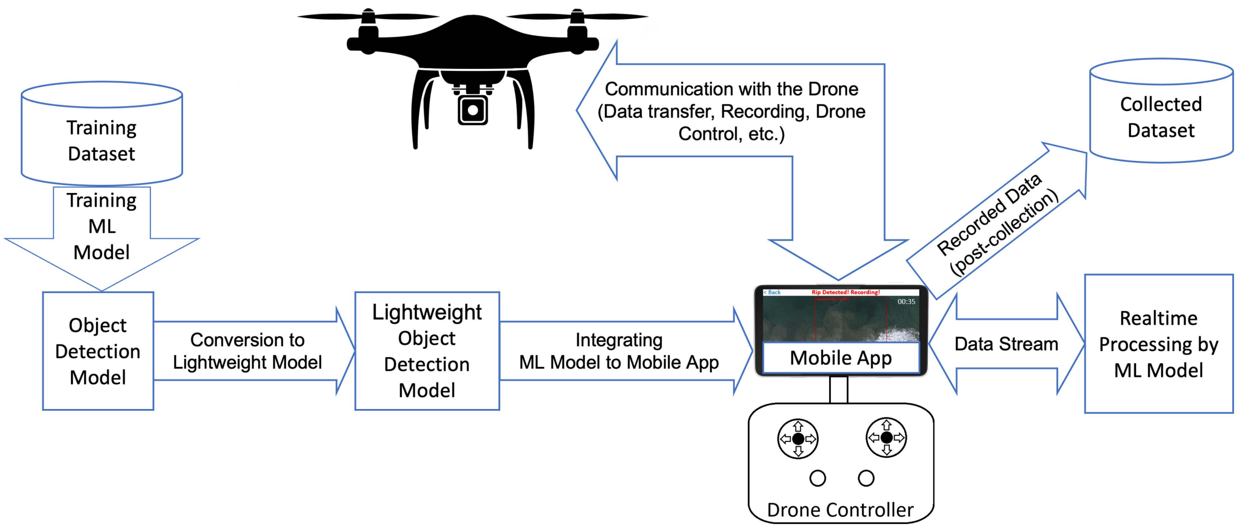

Fig. 1. High level architecture of the realtime ML-assisted data collection system using RipScout.

values. This study evaluates rip current vulnerability by using ML models that predict RTR based on parameters, such as wave height, tide range, and sediment fall velocity. By classifying beach types (e.g., reflective, dissipative) and estimating RTR values, it highlights high-risk areas for rip currents on southern Chinese beaches, which have elevated rip current risks. The MARS model outperformed others in accurately predicting RTR, underscoring its potential for enhanced beach management strategies to mitigate rip current hazards.

The models investigated by these latter sets of papers as well as the findings by Mekhalfi et al. [34] narrowed our field to three candidate models namely: SSD-MobileNetV2, YOLOv8, and EfficientDet D2. In the "Evaluation" section, we present comparisons among these models.

III. System Architecture

Our objective is to design a drone-based system capable of realtime rip current detection and efficient data collection. The high-level architecture of our system is presented in Fig. 1, which includes specific hardware components, as well as software components. This section provides a detailed discussion of the necessary hardware components required for our system, such as the drone and mobile device, and the critical software components, including the mobile application and ML model utilized for rip current detection.

A. Devices and Hardware

We have two main hardware components: a drone with an integrated camera and an associated physical controller, and a mobile device with a touchscreen display physically connected to the controller using a data cable.

1) Drone Hardware Selection: We used a DJI Phantom 4 Pro V2.0 quadcopter, a relatively low-cost professional drone. This model was selected because of our design goal to be *widely deployable*. DJI has the largest share of the drone market in the world and offers a wide range of models suitable for various purposes [36], [47]. The specific model we chose has many advanced features useful for conducting data collection missions, such as improved electronic speed controllers, expanded flight autonomy, and obstacle avoidance using a multicamera and infrared sensing system. It has an onboard gimbal camera featuring a 1-inch 20MP CMOS sensor and a mechanical shutter, eliminating rolling shutter distortion. The camera has a 70° field of view, making it a good choice for computer vision tasks.

The drone is controlled by a physical remote controller that receives a live video feed from the drone camera at a maximum resolution of 1920×1080 and 30 FPS [36]. We use this feed as input for our ML model, while the camera records 4k resolution videos to an onboard microSD card for later scientific analysis. While we showcase our work using the DJI Phantom 4 Pro V2.0, it is important to note that our system is built using the DJI Mobile software development kit (SDK) [36], making it compatible with all DJI drones. In addition, our code is open-source and can be adapted to work with drones from other manufacturers that provide a similar SDK [48], [49]. This makes our system highly adaptable and customizable.

During flights over the beach, drones can be affected by variable wind forces and directions, which may impact the quality of the video feed, resulting in shaky and unstable image frames. These issues can negatively affect the realtime rip current detection process. To address this, we carefully selected our drone with a higher wind resistance rating (22 mph) [36]. However, the system's key component for addressing this issue is its advanced image stabilization, featuring a high-quality three-axis gimbal that isolates the camera from vibrations caused by wind, turbulence, and drone movement. Onboard sensors (GPS, gyroscopes, etc.) detect motion, allowing the gimbal to adjust in real time for stable footage. In addition, electronic image stabilization algorithms [50] further reduce minor shakes and vibrations. All these features can be fine-tuned using the SDK. These capabilities are commonly available in most modern professional drones, making it easier to port RipScout to other platforms [48], [49].

2) Mobile Devices: The physical drone controller has the option of using a smartphone or a tablet to show the live camera feed and other system status. Either of these options would satisfy our need for a *portable* system. However, we also require *realtime* and *accurate* rip current detection. Thus, we opted to use a smartphone as it provides adequate processing power to run the ML models and process videos from drone camera in realtime. We tested our system with an iPhone 12 Pro. While much less powerful than a desktop workstation or server, the A14 Bionic chip on the iPhone 12 Pro is relatively powerful for a mobile device.

One alternate design choice would have been to deploy ML rip current detection directly on the drone. However, one of the major constraints of using drones is the short flight time due to the limited battery power. For the drone we used, each battery lasts a maximum of 30 min [36] with 15–25 min being a more realistic figure. Delegating detection to a separate mobile device prevents the ML module from depleting battery power and reducing flight time.

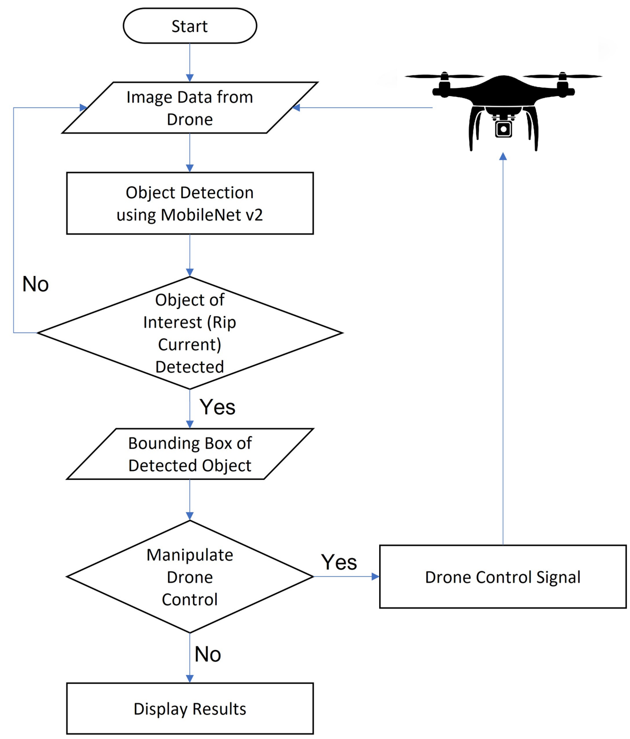

Fig. 2. Diagram of per frame processing by the mobile app.

B. Software Components

We implemented the software components of our system as a mobile application. This app includes the graphical user interface (GUI), ML model for rip current detection, and drone control and communication.

The app receives the live video feed from the drone through the Mobile SDK routines. The video feed serves as input for the ML detection module, which produces detection output in the form of bounding boxes. Upon detecting rip currents, the application initiates specific drone control commands to capture relevant scientific data. Fig. 2 shows the flowchart of the software process.

1) Drone App: The system functionality is presented to the user through a GUI. The GUI supports different aspects of a field data collection mission, including flight planning, providing feedback to the operator when a rip is detected, and specifying data collection actions to take when a rip is detected. There is also a button to enable or disable the background ML process when object detection is not necessary, such as during takeoff, landing, or return to home. Fig. 3 shows the GUI of the RipScout app.

The flight planning screen features an interactive map that allows the user to define the flight path by long-pressing on the map and dropping pins to mark waypoints. In addition, the user can specify the altitude for each waypoint on this screen. Moving to the next screen, the user is presented with options to select the type of data collection action to be performed when a rip current is detected. These actions may include recording videos of predefined lengths from one or more fixed altitudes, capturing footage from different camera angles along circular paths around the detected rip current, or a combination of these options for each detected rip current. Upon starting the mission on the subsequent screen, the user is provided with a live camera feed. Throughout the flight, if a rip current is detected, the locations of these rip currents are indicated by bounding boxes displayed on the live video feed. Furthermore, this screen also shows the status of the ongoing intelligent data collection activities.

Collecting data from various altitudes, angles, and viewpoints of a rip current through circular flights involves three distinct processes. First, the ML model running on the connected mobile device needs to detect the rip current. Second, once the location is identified, the mobile app sends the drone controller the circular flight path information with the detected rip current location at the center, along with user-defined parameters for radius and altitude. Third, as the camera operates independently with its singular gimbal axis, a separate set of camera control data is transmitted to the drone to specify the camera angle and drone heading, ensuring it captures video footage in the correct direction. Note that although many mobile apps for controlling drones offer features, such as “follow me” or “circle around,” these functionalities are primarily designed for common objects, such as cars, bicycles, or individuals. Detection and tracking of rip currents are not supported.

2) ML Model Selection for Rip Current Detection: The criteria for selecting ML models for this application focused on lightweight object detection models capable of detecting rip currents in real-time with dependable accuracy using the limited computational resources of mobile devices. Specifically, we prioritized models that offered a balance between high detection accuracy and computational efficiency, ensuring they could operate within the limited processing power and battery constraints typical of UAV platforms. In addition, these models were chosen based on their proven effectiveness in other UAV and mobile-based detection applications, making them well-suited to our specific deployment environment. We considered only

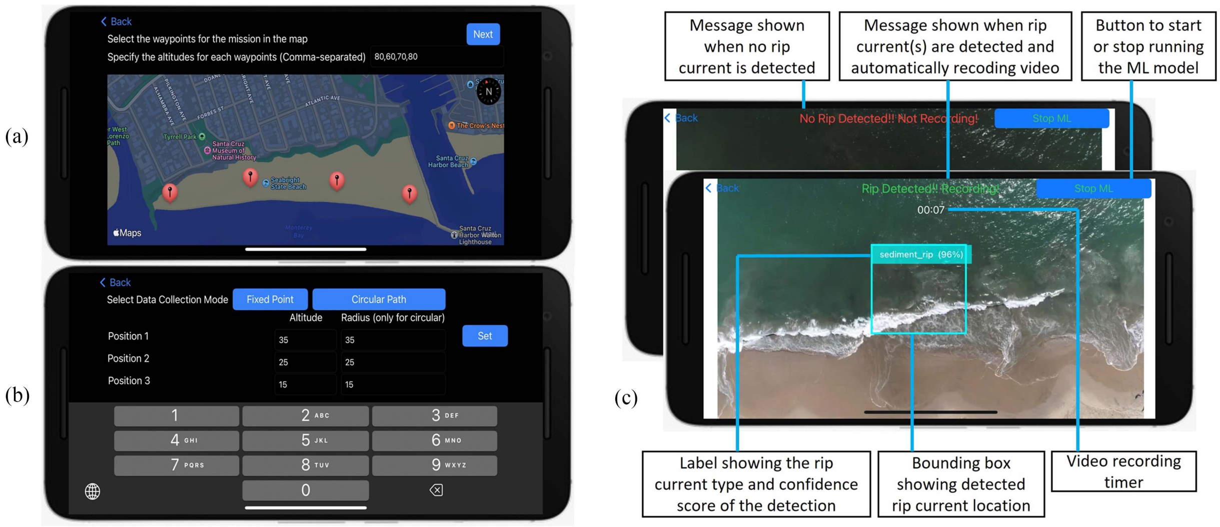

Fig. 3. GUI for planning data collection mission. (a) Interactive map interface for selecting flight plans. (b) Interface to define data collection action plan. (c) Visualization of detected rip current from the live video feed.

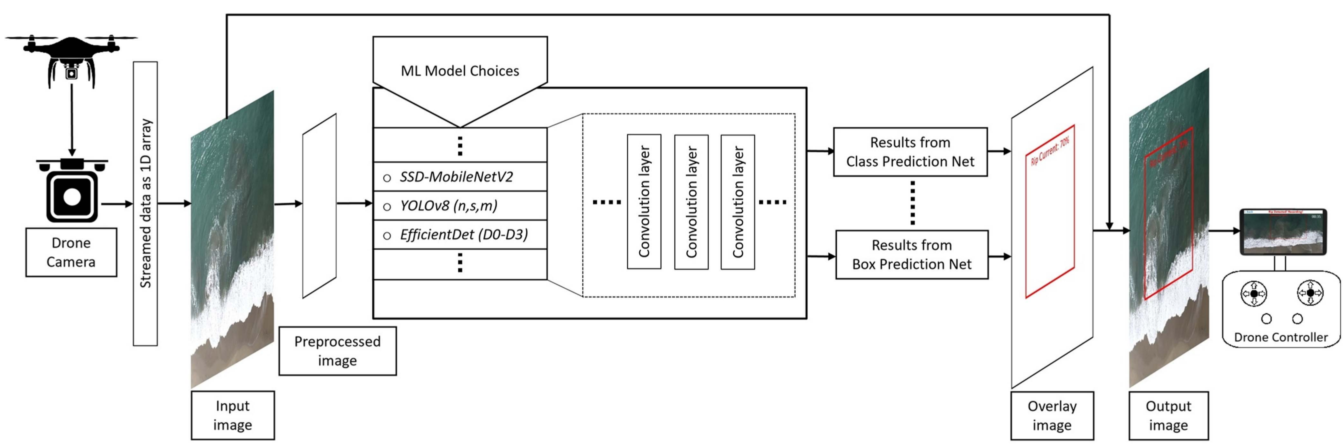

Fig. 4. Pipeline showing how drone video is processed by (choice of) ML model to generate detections in realtime.

models that process each frame of the video data independently to detect rip currents based on their visual signatures. Models that rely on temporal information across frames, such as RipViz, which extracts flow information as vector fields using temporal data, were excluded from consideration [11]. This approach also improves the system's robustness against occasional blurry frames caused by sudden drone movements. To verify that the selected models met our performance requirements, we conducted a comparative analysis, as presented in the "Evaluation" section, which confirmed that the chosen models provided an optimal balance of accuracy, processing speed, and model size.

The rip detector is designed as a modular, plug-and-play component where different ML models can be easily substituted. We selected three models that met the above-mentioned criteria and, based on our literature review, represent three families of state-of-the-art models deployable on mobile devices. As a result, newer and more accurate models from each family can be integrated into our system with minimal or no modifications. Video from the drone is transmitted as a one-dimensional array and converted to image frame buffers as input for the CNN of these ML models. Fig. 4 shows a general architecture of these models. High-level features extracted from the CNN are further processed to generate the detection results, i.e., bounding boxes, class labels, and confidence scores. An overlay image is then generated using the detection results and superimposed over the input image to generate the output image.

IV. Datasets

The datasets associated with this work can be classified into two distinct categories: 1) the initial dataset manually collected for training the rip current detection models, and 2) the data automatically collected by RipScout for subsequent scientific analysis and various other applications.

A. Training Data

There are several mechanisms that give rise to rip currents resulting in different types of rip currents [6]. For an ML model

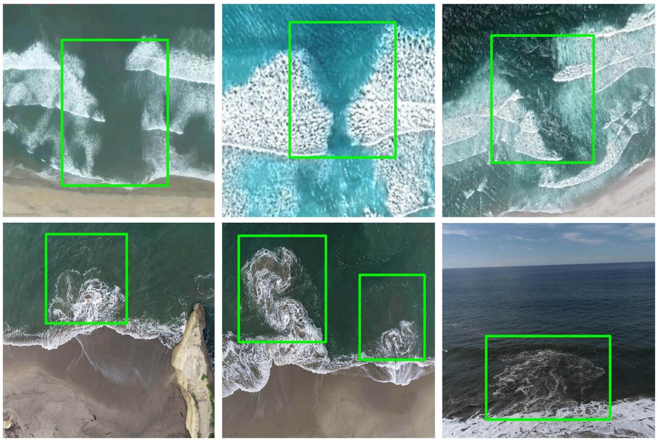

Fig. 5. Some examples labeled images from the dataset with two types of rip currents: channel rips (left three images) and sediment rips (right three images). These images demonstrate the distinct visual characteristics of each type, highlighting the need for separate datasets to train an accurate model.

to detect rip currents, it needs to be trained with many different examples of their visual signature. The models we investigated were designed to detect two types of rip currents. Distinction between rip current types is based on visual appearance, which in itself does not completely indicate the underlying morphological or hydrodynamic mechanisms causing a rip current. The first type considered in this article is visually characterized by a darker, calmer region of water flanked by brighter breaking waves [10]. We shall refer to this type of rip as *channel rips*. This type of rips are typically associated with bathymetry-controlled rip currents [6]. The second type is visually characterized by discolorations caused by sediment-laden water as they are carried away from shore, often forming a plume that extends beyond the breaking waves. We shall refer to this type of rip as *sediment rips*. Sediment rips are typically associated with transient or hydrodynamically controlled rip currents [6]. It is not the goal of this article to classify detected rips based on how a rip was formed but rather how the rip appears visually. It is also important to note that there are rips that have different visual characteristics aside from the two that are studied in this article. For example, presence of rips can be indicated by white foam or water heading offshore. Taking spatial context into consideration, rips can also be indicated based on the wave direction and water texture next to natural and manmade structures. These other types of rips are not considered in this article. In terms of performance of ML detection, each type of rip is considered a different class. Usually, accuracy will drop slightly as more classes are considered. Hence, having more than one class allows us to study how well different ML models perform when there is one, or when there are two classes.

To ensure that our model could accurately detect both types of rip currents, we recognized the need for a separate dataset for sediment rips, in addition to the existing dataset of channel rip images from [10]. As shown in Fig. 5, there are significant visual differences between these two types of rips. These differences are so pronounced that a model trained on one type of rip data would be unable to detect the other type.

Fig. 6. Sediment rips are more obvious from a higher elevation (left) than lower elevation (right).

However, obtaining data for sediment rips is a challenging task, and no existing datasets are available. Hence, we collected a new dataset of sediment rip images by manually flying our drone over the water surface along the shoreline and recording videos. This involved capturing data at different times of day and in various weather conditions, as the appearance of sediment plumes can vary depending on environmental factors. For creating the dataset, we extracted one frame per second from the drone video and manually labeled the frames that contained sediment rip currents. As the visual characteristics of a sediment rip involve water discoloration due to sediment plumes, we identified and labeled the frames that have this visual signature. In some cases, a single frame contained multiple sediment rips, which we labeled accordingly (see Fig. 5, left three images).

We generated a new dataset of 2555 labeled images selectively extracted from 73 videos collected using the drone that captured the visual signature of sediment rip currents. This dataset was combined with the existing dataset of 1780 channel rip images from Silva et al. [10] to train our model to detect both types of rip currents. To mitigate overfitting, we implemented several strategies. The dataset was divided into an 80:20 ratio for training and testing, with 80% of the images allocated for training and 20% for testing. This split ratio is commonly used in ML [51]. We further addressed overfitting by using data augmentation techniques and five-fold cross-validation, which enabled validation across multiple subsets of the data, ensuring robustness and reducing the likelihood of overfitting. Examples of the resulting datasets are displayed in Fig. 5. We have uploaded the sediment rip dataset to a public repository. This will allow other researchers and practitioners to use our dataset to train their models and improve the accuracy of rip current detection.

B. Automated Data Collection

While channel rips can be identified by darker channels of calm water flanked by breaking waves even from ground level, sediment rips are much harder to identify from a lower elevation (see Fig. 6). An ML model trained with the manually collected sediment rip data described previously was used with RipScout to automatically collect new data of sediment rips from different angles and elevations (see Fig. 7).

Such a dataset not only serves purposes for beach monitoring, lifeguard support, and validating rip forecast models, e.g., Dusek and Seim's model [7], but can also be utilized to train models capable of detecting rips observed from lower elevations. Models that are trained on datasets with multiple vantage points can be

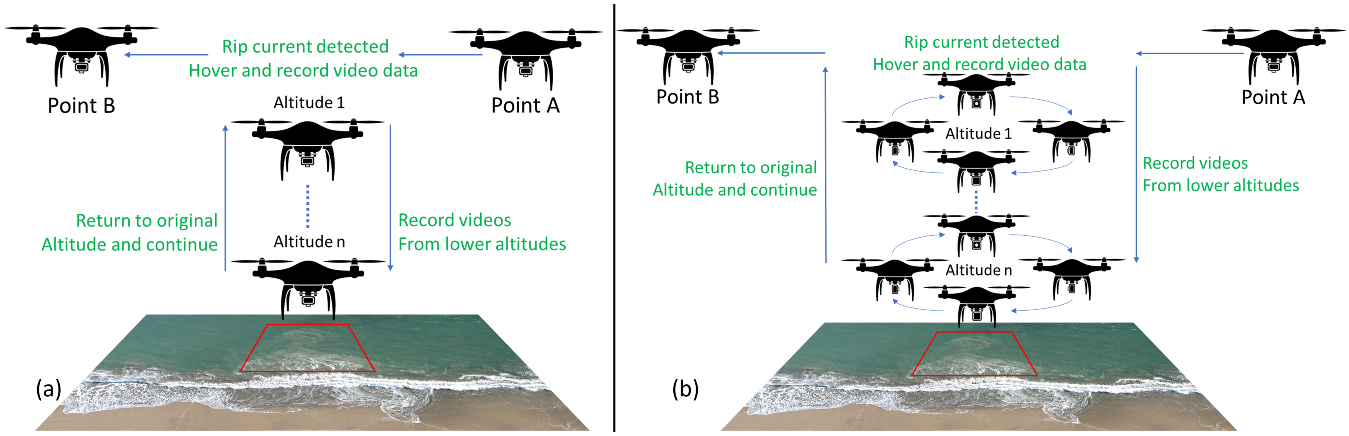

Fig. 7. When RipScout detects a rip between two waypoints, it can be programmed to perform a combination of data collection actions, such as (a) hover in place and record a video of prespecified duration, or go to user specified height before hovering in place and recording the video, then resuming flight from its original height, (b) record a video from user specified radius and height with the detection location as the center.

employed for rip current detection on lower elevation platforms, such as web cameras and smartphones.

V. Field Testing

We collected data and conducted extensive field tests of our system on six public beaches in California. Our field tests were meant to validate the suitability of RipScout for automated data collection. Each field test consisted of multiple drone flights, all conducted by a drone pilot certified by the Federal Aviation Administration (FAA) under the Small UAS Rule (Part 107), with the drone properly registered with the FAA [52]. We obtained permits from federal and state authorities, including the California State Park System and the Monterey Bay Marine Sanctuary, to ensure compliance with all relevant regulations. Over a period of 18 months, from January 2022 to June 2023, we performed a total of 36 drone flight missions to collect training data and test our system in real-world conditions. All flights were conducted in strict adherence to local and federal regulations and FAA guidelines [52]. Our field tests were conducted in compliance with the approval and guidelines provided by the Office of Research Compliance Administration at the University of California, Santa Cruz.

The field tests were conducted with careful consideration of varying environmental conditions that could affect rip current visibility and detection accuracy. We used a range of tools, such as tide and weather monitoring apps and NOAA forecast tools, to monitor conditions before planning each field trip [7], [53]. These conditions included weather elements, such as sunlight and cloud cover, as well as sea state factors, such as wave height, tide levels, and water clarity. On clear, sunny days with minimal cloud cover, rip currents were more easily detectable due to optimal lighting and visibility, allowing the ML models to perform more accurately. In contrast, cloudy conditions and periods of glare from direct sunlight posed challenges by creating shadows and reflections on the water, occasionally resulting in false negatives and false positives.

Although the drone has good wind resistance and image stabilization capabilities, even the most advanced drones may struggle in extremely strong winds or gusts. The drone can detect such extreme wind conditions and alert the user through the mobile app, either suggesting the user take action or automatically returning to the landing pad for safety. Sea state also played a crucial role. For instance, higher wave heights and active surf conditions helped accentuate rip currents, as the calmer, channel-like structure of the rips contrasted against breaking waves, improving detection reliability. Meanwhile, rip currents were more challenging to identify on days with low wave activity, especially sediment rips, which are visually less distinct without wave movement. These variabilities underline the adaptability of the RipScout system under different conditions and highlight the challenges of achieving consistent detection accuracy across diverse beach environments.

Fig. 7 illustrates a typical data collection scenario when using the system in the field. The drone flies from Point A to Point B following a predefined flight plan. During the flight, the ML model runs on the mobile device connected to the controller and processes the live video feed from the drone camera. If a rip current is detected, the drone automatically stops and performs a data collection activity, such as recording a video clip for 30 s at several altitudes or flying in a circle to capture the rip from multiple angles. When at least one rip current is present, the app is instructed to send control information to the drone to capture a high-quality 4 K video of a predefined length and save it on the onboard microSD card. The quality of this recorded video is much higher than the quality of the live feed used for realtime ML processing, as it is intended for further analysis, research, and archival purposes. The location of the rip current is shown on the app's preview screen using bounding boxes. Once the data capture is completed, the drone returns to its original flight plan. The drone pilot always has full control over the drone and can manually override such programmed behavior at any time.

For four of our field tests, we explicitly compared the performance of our ML-assisted system with the performance of drone pilots on an unaugmented drone. A total of ten drone pilots, who are not experts in rip current detection, participated in these comparison trials. The ten participants were recruited by snowball sampling through our network in academia, and informed consent was obtained from them. The field tests were performed at a state beach in California where sediment rip currents are

TABLE I
COMPARISON OF DETECTION ACCURACY, MODEL SIZE, AND PROCESSING SPEED OF THREE ML MODELS WHEN TRAINED ON A SINGLE CLASS: EITHER CHANNEL RIP OR SEDIMENT RIP

ML model	SSD-MobileNetV2	YOLOv8m	EfficientDet D2
Accuracy (channel rip)	82.1%	87.3%	94.8%
Accuracy (sediment rip)	76.06%	92.6%	92.9%
Saved weight of ML model (Megabytes)	4.5	49.6	7.0
Training time (channel rip)	10 h	4 h	5 h
Training time (sediment rip)	9 h	3.5 h	5 h
Processing speed in frame per second	33	25	17

TABLE II
COMPARISON OF DETECTION ACCURACY, MODEL SIZE, AND PROCESSING SPEED OF THREE ML MODELS WHEN TRAINED TO DETECT AND DISTINGUISH BETWEEN TWO CLASSES: EITHER CHANNEL RIP OR SEDIMENT RIP

ML model	SSD-MobileNetV2	YOLOv8m	EfficientDet D2
Average accuracy	78.45%	88.3%	93.1%
Accuracy (channel rip)	80.83%	84.8%	93.6%
Accuracy (sediment rip)	76.06%	91.8%	92.6%
Saved weight of ML model (Megabytes)	4.5	49.6	7.0
Training time	14 h	5 h	8 h
Processing speed in frame per second	33	25	17

prevalent to provide enough data samples for comparison. In each case, days and times were decided based on weather, tidal, and wave conditions in which rip currents were likely to be present. Before the field trips, participants were provided with tutorials on how to detect rip currents. The drone's flight path was preprogrammed to fly through a set of waypoints at a predefined altitude overlooking the shoreline. At the beginning of each field test, an initial flight pass was performed to collect an overview of the location for future review by rip current experts. The next round of flight followed the same path while running RipScout. Guided by the ML, every time a rip current was detected, the drone stopped and recorded a high-resolution video. In subsequent rounds, participants flew the drone manually through the same trajectory, and every time they visually determined the presence of a rip current, they manually triggered recording a high-resolution video. RipScout and participants were compared in terms of accuracy in locating rip currents, as well as the rate at which data samples could be collected.

VI. EVALUATION

We analyze two critical aspects of our system. First, we compare the accuracy, processing speed, and resource utilization of three ML models for rip current detection. Second, we compare the efficiency of our system for collecting rip current data via a series of field tests using drones with and without ML support.

A. ML Model Performance

We analyze the detection accuracy of the three models when tasked with detecting one class of object only (either channel rips or sediment rips) in Table I. We also analyzed the impact on detection accuracy when the three models are tasked with detecting rips from one of two classes in Table II. For all comparisons, models were trained on a desktop with a GPU, and tested on CPU-based implementation running on an iPhone 12 Pro. Conversion of an implementation using a GPU to a CPU implementation to run on a smartphone involves conversion to a FlatBuffer format [54]. All the detection accuracy figures are based on our implementation of the three ML models. Training and testing were done on the same dataset described in the "Datasets" section.

Tables I and II present a comparative analysis of three ML models—SSD-MobileNetV2, YOLOv8m, and EfficientDet D2—evaluated for their performance in rip current detection. In Table I, EfficientDet D2 achieved the highest detection accuracy for both channel and sediment rips, with an accuracy of 94.8 % and 92.9 %, respectively, which underscores its suitability for detecting both types of rip currents under single-class training conditions. YOLOv8m demonstrated competitive accuracy, especially for sediment rips, with 92.6 %, although it ran at a slightly slower frame rate (25 fps) compared to SSD-MobileNetV2. SSD-MobileNetV2, while achieving the highest processing speed of 33 fps, showed lower accuracy scores, making it a less optimal choice for high-precision applications.

Table II expands on this comparison by evaluating each model's performance on multiclass detection (distinguishing between both channel and sediment rips). In this scenario, EfficientDet D2 maintained robust accuracy (93.1 %) across both classes, demonstrating superior balance between accuracy and processing speed. Although YOLOv8m performed well, achieving 88.3 % average accuracy, its larger model size (49.6 MB) could be a drawback on memory-constrained mobile devices. SSD-MobileNetV2's accuracy dropped further with multiclass detection, reinforcing its tradeoff between speed and detection precision.

These tables highlight EfficientDet D2 as the most balanced model for both accuracy and practical deployment on UAVs, particularly given its acceptable processing speed and smaller model size compared to YOLOv8m. This choice supports RipScout's goal of real-time, reliable rip current detection in resource-constrained environments.

We were not able to exactly replicate the detection accuracy for SSD-MobileNetV2 as reported by Rampal et al. [16], likely due to the conversion to CPU implementation and differences in

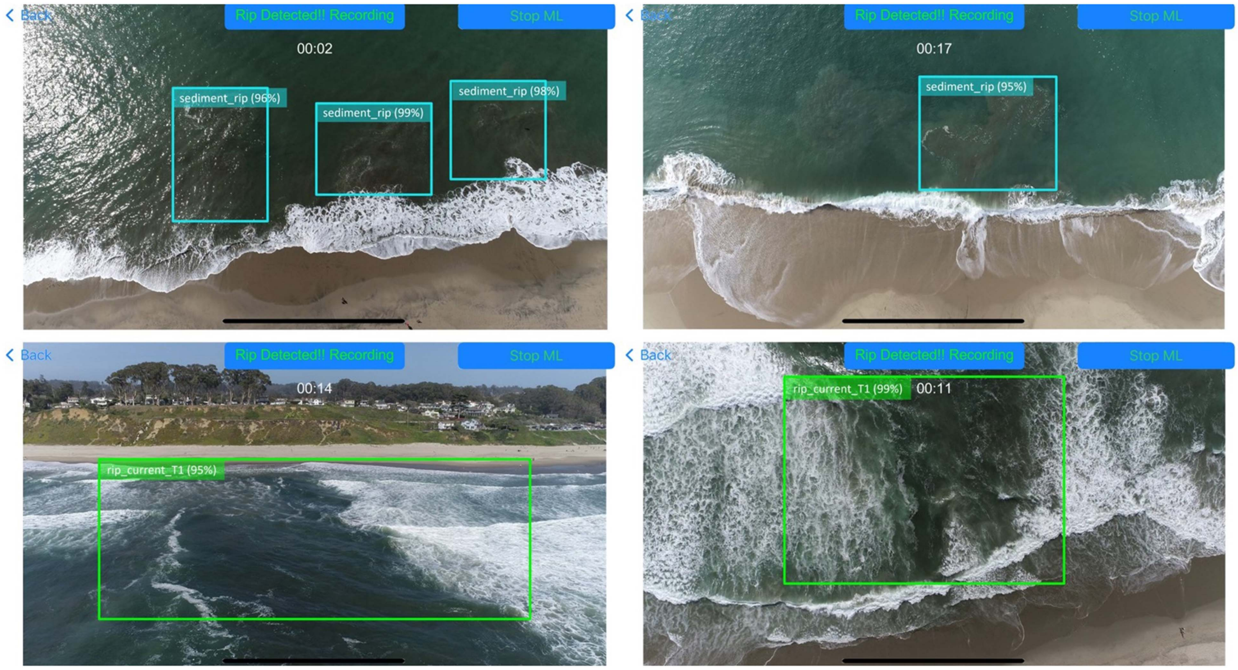

Fig. 8. Here are some examples of realtime automatic detections of two types of rip currents using RipScout: sediment rips (top row) and channel rips (bottom row). As shown in the top-left image, RipScout can detect multiple rip currents in a single frame. The two images in the bottom row demonstrate that RipScout can detect rip currents from both side and top views.

dataset (no code or data were shared). Likewise, the detection accuracy for Dumitriu et al. [12] are different as they used YOLOv8 for segmentation while we used it for detection. In addition, a different accuracy metric was used. Among the five variations of YOLOv8, we selected YOLOv8m based on the benchmark performed by Dumitriu et al. [12]. There are eight variants of EfficientDet to select from, starting with EfficientDet D0 with expected input size 512×512 to EfficientDet D7 with expected input size 1536×1536. We selected EfficientDet D2 as it takes 768×768 as input, the closest to the 1280×720 resolution video transmitted from the drone while providing realtime processing speed.

Looking at Table I, we see that EfficientDet D2 has the highest detection accuracy for detecting channel rips (when trained with channel rip dataset). EfficientDet D2 also narrowly edged out YOLOv8m in detecting sediment rips (when trained with sediment rip dataset). Note that model sizes are all reasonably small to run on most smartphone. EfficientDet D2 runs approximately about half as fast as SSD-MobileNetV2, but is still acceptable for realtime detection when every other frame from the drone is dropped.

As expected, all three methods incurred a slight drop in detection accuracy when a second class of objects was added (see Table II). This can be seen when comparing the detection accuracy for channel rips amongst the three models in Tables I and II. The same is true for sediment rips. The only exception is SSD-MobileNetV2 for sediment rips where accuracy remains unchanged. Since each of the three different models was presented with an equal number of test cases for channel and sediment rips, the average accuracy is the simple average of the detection accuracy for each type of rip. The top performer in terms of accuracy EfficientDet D2. Its model size and processing speed are within the requirements for a lightweight mobile app and realtime processing.

Training times are included for completeness. Training was performed using TensorFlow on a desktop machine equipped with an Intel Core(TM) i7 2.60 GHz microprocessor, 16 Gigabyte main memory, and an Nvidia RTX 2070 GPU with 8 Gigabyte video memory. After completing the training, we converted the models into the optimized FlatBuffer format for integration into both iOS and Android apps.

Fig. 8 showcases the realtime detection of the two types of rip currents by RipScout fitted with an EfficientDet D2 detector. Sediment rips are highlighted in cyan bounding boxes while channel rips are highlighted in green bounding boxes. RipScout can detect multiple rip currents in a single frame, as illustrated by the presence of multiple bounding boxes.

The tables present the accuracy of rip current detection on the test dataset and ground truth published by Silva et al. [10], along with additional video clips containing sediment rips that we collected using the drone. Accuracy is calculated using the same method as Silva et al. [10], based on the object detection metrics proposed by Padilla et al. [55]. These object detection metrics are widely accepted and appropriate for object detection problems in the computer vision field

$$\text{accuracy} = \frac{\text{correct\_labels}}{\text{total\_frames}}.$$

Frames were considered classified as correct if the detected bounding boxes had an intersection over union (IoU) score versus ground truth bounding boxes above 0.3. IoU is calculated

TABLE III
FIELD TEST COMPARISON OF RIP CURRENT DETECTION EFFICIENCY USING DRONES WITH (RIPSCOUT) VERSUS WITHOUT (HUMAN ONLY) THE AID OF ML

Field test	Average time to capture Each rip current (seconds)		Data collection speed improved (times faster)
	Human	RipScout	
1	118.50	29.60	4.00
2	148.17	34.00	4.36
3	102.71	32.00	3.21
4	179.60	42.00	4.28
Overall	137.24	34.40	3.99

as follows:

$$\text{IoU} = \frac{\text{area\_of\_intersection}}{\text{area\_of\_union}}.$$

For ML models trained to detect two classes of rip currents (or more in future work), mean intersection over union (mIoU) was calculated. mIoU is the average IoU across all classes or instances in a dataset, calculated as follows:

$$\text{mIoU} = \frac{1}{C}\sum_{i=1}^{C}\frac{\text{area\_of\_intersection}}{\text{area\_of\_union}}$$

where C is the total number of classes.

We verified the accuracy of each variation of the ML models by generating recall $\times$ precision curves using the open-source toolbox provided by Padilla et al. [55]. Apart from evaluating the accuracy of each model, we also assessed their memory storage requirements and average processing time per frame. Our findings indicate that all three models are suitable for realtime processing and have memory requirements close to the average iOS size of 35 MB.

B. Efficiency of RipScout

In this section, we compare the efficiency of data collection using RipScout fitted with EfficientDet D2 to traditional data collection involving human participants without ML assistance. Our comparison is based on several field tests, each involving multiple drone flights as described in the "Field Testing" section. We only considered data from a field test if there were more than one rip current present at that time. Table III presents a summary and comparison of the average time required to capture each rip current by RipScout and human participants. Our results show that RipScout had a significantly lower average time of 34.40 s to capture a rip current compared to 137.24 s for human participants. In addition, the overall flight time required by human participants was approximately four times greater than that of RipScout. This finding indicates that the use of RipScout can provide more coverage and/or extended monitoring. This is noteworthy given that the flight time of each battery is approximately 15–25 min depending on weather conditions and operational use (e.g., whether recording or not).

While our field tests utilized the EfficientDet D2 model, one could also replace it with other models such as those presented in Tables I and II. We would expect similar efficiency gains of ML assisted detection and collection versus humans only, with the corresponding reduction in detection accuracy, or an improvement if a better lightweight model becomes available in the future.

In addition to comparing RipScout with nonexpert human participants, we also evaluated our system and collected feedback from an experienced lifeguard. Lifeguards are much more experienced in detecting rip currents than our other participants. The detection performance of lifeguards averaged 34.2 s per rip, on par with that of RipScout.

In terms of accuracy, RipScout was also able to detect all the rips that the lifeguard expert found. Although we only have one expert validation point at this time, the accuracy result was reassuring that RipScout performed satisfactorily.

C. Accuracy in the Field Tests

The accuracy metric presented in Tables I and II ignores false positives (rips that are not present but flagged as present) and false negatives (rips that are present but not detected). We looked for false positives and false negatives on the videos from the field trip with the help of a rip current expert. In our analysis, we found that human participants had a 31% false positive rate, whereas RipScout had a 17% false positive rate. On the other hand, there were no general instances of false negatives for both human participants and RipScout. For the human participants, we believe the nature of the task (i.e., asking them to look for rip currents) led them to be more cautious and erred on flagging something as a rip when it is not. Hence, false negative rate was negligible. For RipScout, indeed there are frames where a rip was present but was not detected. However, at our sampling rate of 30 FPS, the same rip would always be detected in other frames, but necessarily all the frames. Hence, we marked the rip as being detected, leading to no false negative detection of the rip.

For the purpose of using RipScout for automated data collection, we considered the 17% false positive rate acceptable since we want to gather as much data as possible during each drone flight. The collected data can later be cleaned up using human experts (e.g., relabel the 17% false positive detections) or with a higher accuracy and more compute intensive two-stage ML model.

Moreover, we can further reduce the false positive and false negative rate by employing common ML model optimization techniques such as data augmentation, increasing the amount and quality of training data, and tuning the model's parameters. These techniques are widely used in the field of ML to improve a deep learning models' accuracy continuously [56].

VII. CONCLUSION

In summary, RipScout is a system for rip current data collection which integrates ML rip current detection into the flight control process. The ML system is carefully designed to work well under limited computational constraints, and we show that rip current detection accuracy of EfficientDet D2 is almost 5% better than its closest rival. Actual field tests show that this system is effective, allowing data to be collected almost four times faster than without RipScout, alleviating the need for domain experts to be present with the drone operator. Furthermore, with further validation, RipScout can be used to improve beach safety by providing rapid and wide coverage of beach monitoring for rips, helping to reduce fatigue of lifeguarding operations, especially in communities where they are understaffed.

RipScout highlights the locations of rips with bounding boxes. This not only helped users identify rip currents quickly, but it also served as an effective tool for learning. We observed that after using the system, users improved their rip current detection skills even without the assistance of the ML model. This suggests that the system can serve as a training tool for beachgoers to learn how to spot rip currents and improve their overall beach safety awareness.

During our field tests, we observed that RipScout can readily detect rip currents in bright daylight, where it is difficult for the human eyes to spot rip currents due to insufficient brightness and the small display of mobile devices. This helps make data collection less error prone. However, in poor lighting conditions, including dense fog or extreme glare in the video stream from the drone, the ML detection model will have difficulty finding rips. Our future work will focus on further enhancing the detection accuracy of RipScout by augmenting our training data with additional data obtained in such diverse conditions, as well as using generative ML models to add these effects to our existing training data. Likewise, any ML detection model can only recognize rips that they were trained on. We plan to collect training data on rip currents with other types of visual signatures to further enhance RipScout.

Another future direction involves adapting RipScout to detect other beach hazards or similar aquatic phenomena, broadening its utility. Beyond rip currents, the system architecture could be generalized for applications requiring large-area searches, such as biodiversity monitoring or search and rescue operations. We also plan to extend RipScout to support a wide variety of UAVs, including custom-made drones, which would enable customized processing platforms, vision sensors, battery technology, and energy-efficient model designs to further extend flight time for more comprehensive monitoring.

VIII. Data Availability

To encourage the use of this system for rip current data collection and other applications, the software code and all specifications are available as an open-source project at.[1] To encourage research on rip currents using visual data, our datasets, including the new dataset of 2555 frames of sediment rip currents, are also available in the same repository.

Acknowledgment

The statements, findings, conclusions, and recommendations are those of the authors and do not necessarily reflect the views of SECOORA, NOAA, or UCSC. The content of the information provided in this publication does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. The authors thank the California State Parks System and the Monterey Bay National Marine Sanctuary for their support and the permissions granted to operate their drone for this and related research. The authors also thank the Santa Cruz Port District, the California State Park Lifeguards, Nick Carver, and the other field test participants for their invaluable assistance in evaluating our system and providing feedback to help us improve RipScout. Finally, the authors acknowledge support from the Google Cloud Research Credits program and AWS Cloud Credit for Research.

[1] https://github.com/fahimhkhan/ripscout

References

[1] Y. Zhang, W. Huang, X. Liu, C. Zhang, G. Xu, and B. Wang, "Rip current hazard at coastal recreational beaches in China," *Ocean Coastal Manage.*, vol. 210, 2021, Art. no. 105734.

[2] A. Retnowati, M. A. Marfai, and J. S. Sumantyo, "Rip currents signatures zone detection on alos palsar image at parangtritis beach, Indonesia," *Indonesian J. Geogr.*, vol. 43, no. 2, pp. 12–27, 2012.

[3] L. Mucerino, L. Carpi, C. F. Schiaffino, E. Pranzini, E. Sessa, and M. Ferrari, "Rip current hazard assessment on a sandy beach in Liguria, NW Mediterranean," *Natural Hazards*, vol. 105, pp. 137–156, 2021.

[4] V. A. Gensini and W. S. Ashley, "An examination of rip current fatalities in the United States," *Natural Hazards*, vol. 54, no. 1, pp. 159–175, Jul. 2010. [Online]. Available: https://doi.org/10.1007/s11069-009-9458-0

[5] A. de Silva, *Deep Learning for Flow Feature Detection*. Berkeley, CA, USA: Univ. California, Santa Cruz, 2023.

[6] B. Castelle, T. Scott, R. Brander, and R. McCarroll, "Rip current types, circulation and hazard," *Earth- Sci. Rev.*, vol. 163, pp. 1–21, 2016.

[7] G. Dusek and H. Seim, "A probabilistic rip current forecast model," *J. Coastal Res.*, vol. 29, no. 4, pp. 909–925, 2013. [Online]. Available: https://doi.org/10.2112/JCOASTRES-D-12-00118.1

[8] S. B. Leatherman, "Rip current measurements at three South Florida beaches," *J. Coastal Res.*, vol. 33, no. 5, pp. 1228–1234, 2017.

[9] J. MacMahan et al., "An introduction to rip currents based on field observations," *J. Coastal Res.*, vol. 27, no. 4, pp. iii–vi, 2011.

[10] A. de Silva, I. Mori, G. Dusek, J. Davis, and A. Pang, "Automated rip current detection with region based convolutional neural networks," *Coastal Eng.*, vol. 166, 2021, Art. no. 103859.

[11] A. de Silva et al., "RipViz: Finding rip currents by learning pathline behavior," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 7, pp. 3930–3944, Jul. 2024.

[12] A. Dumitriu, F. Tatui, F. Miron, R. T. Ionescu, and R. Timofte, "Rip current segmentation: A novel benchmark and YOLOv8 baseline results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2023, pp. 1261–1271.

[13] C. Maryan, M. T. Hoque, C. Michael, E. Ioup, and M. Abdelguerfi, "Machine learning applications in detecting rip channels from images," *Appl. Soft Comput.*, vol. 78, pp. 84–93, 2019.

[14] I. Mori, A. de Silva, G. Dusek, J. Davis, and A. Pang, "Flow-based rip current detection and visualization," *IEEE Access*, vol. 10, pp. 6483–6495, 2022.

[15] S. Philip and A. Pang, "Detecting and visualizing rip current using optical flow," in *Proc. EuroVis (Short Papers)*, 2016, pp. 19–23.

[16] N. Rampal, T. Shand, A. Wooler, and C. Rautenbach, "Interpretable deep learning applied to rip current detection and localization," *Remote Sens.*, vol. 14, no. 23, 2022, Art. no. 6048. [Online]. Available: https://www.mdpi.com/2072-4292/14/23/6048

[17] A. H. Rashid, I. Razzak, M. Tanveer, and A. Robles-Kelly, "RipDet: A fast and lightweight deep neural network for rip currents detection," in *Proc. 2021 Int. Joint Conf. Neural Netw.*, IEEE, 2021, pp. 1–6.

[18] M. B. Bejiga, A. Zeggada, and F. Melgani, "Convolutional neural networks for near real-time object detection from UAV imagery in avalanche search and rescue operations," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, IEEE, 2016, pp. 693–696.

[19] B. Vergouw, H. Nagel, G. Bondt, and B. Custers, "Drone technology: Types, payloads, applications, frequency spectrum issues and future developments," in *The Future of Drone Use*. Berlin, Germany: Springer, 2016, pp. 21–45.

[20] R. Merkert and J. Bushell, "Managing the drone revolution: A systematic literature review into the current use of airborne drones and future strategic directions for their effective control," *J. Air Transport Manage.*, vol. 89, 2020, Art. no. 101929.

[21] Y. Cao et al., "VisDrone-DET2021: The vision meets drone object detection challenge results," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2847–2854.

[22] D. Du et al., "VisDrone-DET2019: The vision meets drone object detection in image challenge results," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019, pp. 213–226.

[23] H. Fan et al., "VisDrone-MOT2020: The vision meets drone multiple object tracking challenge results," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 713–727.

[24] P. Zhu et al., "VisDrone-DET2018: The vision meets drone object detection in image challenge results," in *Proc. Eur. Conf. Comput. Vis.Workshops*, 2018, pp. 437–468.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, vol. 28, pp. 91–99.

[26] J. Lee, J. Wang, D. Crandall, S. Šabanović, and G. Fox, "Real-time, cloud-based object detection for unmanned aerial vehicles," in *Proc. First IEEE Int. Conf. Robotic Comput.*, 2017, pp. 36–43.

[27] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016, pp. 21–37.

[28] J. Huang et al., "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3296–3297.

[29] C. Sun, W. Zhan, J. She, and Y. Zhang, "Object detection from the video taken by drone via convolutional neural networks," *Math. Problems Eng.*, vol. 2020, 2020, Art. no. 4013647.

[30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[31] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 2020, pp. 10778–10787.

[32] G. Jocher et al., "ultralytics/yolov5: v7.0 - YOLOv5 SOTA realtime instance segmentation," *Zenodo*, 2022.

[33] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 213–229.

[34] M. L. Mekhalfi, C. Nicolò, Y. Bazi, M. M. A. Rahhal, N. A. Alsharif, and E. A. Maghayreh, "Contrasting YOLOv5, transformer, and EfficientDet detectors for crop circle detection in desert," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 3003205.

[35] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extract.*, vol. 5, no. 4.

[36] DJI, "Phantom 4 pro v2.0 - specifications - dji," 2023. [Online]. Available: https://www.dji.com/phantom-4-pro-v2/specs

[37] C. Chen et al., "RRNet: A hybrid detector for object detection in drone-captured images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.Workshops*, Oct. 2019, pp. 100–108.

[38] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2778–2788.

[39] P. Nousi, I. Mademlis, I. Karakostas, A. Tefas, and I. Pitas, "Embedded UAV real-time visual object detection and tracking," in *Proc. IEEE Int. Conf. Real-time Comput. Robot.*, IEEE, 2019, pp. 708–713.

[40] X. Wu, W. Li, D. Hong, R. Tao, and Q. Du, "Deep learning for unmanned aerial vehicle-based object detection and tracking: A survey," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 1, pp. 91–124, Mar. 2022.

[41] A. Rohan, M. Rabah, and S.-H. Kim, "Convolutional neural network-based real-time object detection and tracking for parrot AR drone 2," *IEEE Access*, vol. 7, pp. 69575–69584, 2019.

[42] J. Deng, Z. Shi, and C. Zhuo, "Energy-efficient real-time UAV object detection on embedded platforms," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 39, no. 10, pp. 3123–3127, Oct. 2020.

[43] N. Tijtgat, W. Van Ranst, T. Goedeme, B. Volckaert, and F. De Turck, "Embedded real-time object detection for a UAV warning system," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 2110–2118.

[44] M. Vajgl, P. Hurtik, and P. Števuliáková, "Drone real-time control based on pattern matching," in *Proc. Joint 17th World Congr. Int. Fuzzy Syst. Assoc. 9th Int. Conf. Soft Comput. Intell. Syst.*, IEEE, 2017, pp. 1–6.

[45] D. Zhu, R. Qi, P. Hu, Q. Su, X. Qin, and Z. Li, "YOLO-Rip: A modified lightweight network for rip currents detection," *Front. Mar. Sci.*, vol. 9, 2022, Art. no. 930478. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fmars.2022.930478

[46] M. Najafzadeh, S. Basirian, and Z. Li, "Vulnerability of the rip current phenomenon in marine environments using machine learning models," *Results Eng.*, vol. 21, 2024, Art. no. 101704. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590123023008319

[47] M. Yousef, F. Iqbal, and M. Hussain, "Drone forensics: A detailed analysis of emerging DJI models," in *Proc. 11th Int. Conf. Inf. Commun. Syst.*, IEEE, 2020, pp. 66–71.

[48] P. Developers, "Parrot developer documentation," 2024. [Online]. Available: https://developer.parrot.com/docs

[49] Skydio, "Developer tools - skydio," 2024. [Online]. Available: https://www.skydio.com/developer-tools

[50] C. Morimoto and R. Chellappa, "Evaluation of image stabilization algorithms," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process., ICASSP'98 (Cat. No 98CH36181)*, IEEE, 1998, vol. 5, pp. 2789–2792.

[51] V. R. Joseph, "Optimal ratio for data splitting," *Stat. Anal. Data Mining: ASA Data Sci. J.*, vol. 15, no. 4, pp. 531–538, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.11583

[52] FAA, "FAA unmanned aircraft systems (UAS)," Mar. 2022. [Online]. Available: https://www.faa.gov/uas/

[53] G. Dusek, W. V. Sweet, M. J. Widlansky, P. R. Thompson, and J. J. Marra, "A novel statistical approach to predict seasonal high tide flooding," *Front. Mar. Sci.*, vol. 9, 2022, Art. no. 1073792.

[54] Google, "Flatbuffers: Flatbuffers," 2023. [Online]. Available: https://flatbuffers.dev/

[55] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics*, vol. 10, no. 3, 2021, Art. no. 279. [Online]. Available: https://www.mdpi.com/2079-9292/10/3/279

[56] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT press, 2016.

Fahim Hasan Khan received the B.S. degree in computer science and engineering from the Rajshahi University of Engineering and Technology, Rajshahi, Bangladesh, in 2009, the M.S. degree in computer science from the University of Calgary, Calgary, AB, Canada, in 2018, and the Ph.D. degree in computer science and engineering from the University of California, Santa Cruz, CA, USA, in 2024.

He is currently an Assistant Professor with the Computer Science and Software Engineering Department, California Polytechnic State University, San Luis Obispo, CA, USA. His research interests include computer graphics, computer vision, applied machine learning, and flow visualization.

Donald Stewart received the B.S. degree in computer science, in 2017 and the M.S. degrees in computer science, in 2020 from the University of California, Santa Cruz, CA, USA.

His research interests include computer vision and deep learning, with a focus on applications that enhance public safety.

Akila de Silva received the B.S. in computer science degree from the Asian Institute of Technology, Khlong Nueng, Thailand, the M.S. in computer science degree from Columbia University, New York, NY, USA, in 2017, and the Ph.D. degree in computer science from the University of California at Santa Cruz, Santa Cruz, CA, USA, in 2023.

He is currently an Assistant Professor of computer science with San Francisco State University, San Francisco, CA, USA. His research interests include visualization, machine learning, deep learning, and computer vision.

Ashleigh Palinkas received the B.A. degree in anthropology from Fordham University, New York, NY, USA, in 2009, and the M.A.S. degree in marine biodiversity and conservation from the Scripps Institution of Oceanography, University of California, San Diego, CA, USA, in 2014.

She is currently a Marine Research Associate with California Sea Grant, San Diego, CA, USA, and the University of California, San Diego, San Diego, CA, USA. Her research interests include resilient coastal communities and economies, and sustainable fisheries and aquaculture.

Gregory Dusek received the B.S. degree in applied mathematics and the M.S. degree in teaching and curriculum from the University of Rochester, Rochester, NY, USA, in 2004 and 2005, respectively, and the Ph.D. degree in marine science from the University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, in 2011.

Since 2014, he has been the Chief Scientist with the NOAA National Ocean Service, Center for Operational Oceanographic Products and Services, Silver Spring, MD, USA. He is currently a Physical Oceanographer.

James Davis (Senior Member, IEEE) received the Ph.D. degree in computer science from Stanford University, Stanford, CA, USA, in 2002.

From 2002 to 2004, he was a Senior Research Scientist with Honda Research Institute, San Jose, CA, USA. He is currently a Professor of computer science and engineering with the University of California at Santa Cruz, Santa Cruz, CA, USA. He has authored or coauthored more than 100 peer-reviewed publications, patents, and invited talks. His research interests include both traditional computer science and technology applications to social issues.

Dr. Davis was the recipient of the IEEE ICRA 2003 Best Vision Paper, IEEE ICCV 2009 Marr Prize, and an NSF CAREER Award. His teaching has twice been awarded for innovative style, including a course on the importance of technology to social entrepreneurship. He is on the advisory boards of several for-profit and nonprofit organizations, and co-founded Bellus3D, in 2015.

Alex Pang received the B.S. degree (*magna cum laude*) in industrial engineering from the University of the Philippines, Quezon City, Philippines, in 1981, and the Ph.D. degree in computer science from the University of California, Los Angeles, Los Angeles, CA, USA, in 1990.

He is currently a Professor in computer science and engineering with the University of California at Santa Cruz, Santa Cruz, CA, USA. He has authored or coauthored more than 100 refereed papers in journals and conferences in both scientific and information visualization. His research interests include comparative and uncertainty visualization, flow and tensor visulization, and visualization for social good.