Product Discussion

Thursday October 10, 1:15 - 2:00 PM

Session overview

- Status & limitations of WebCOOS products
- What is working well (& what is not quite there yet)
- Set the stage for siting discussions in science showcase to follow
    - Shoreline change: Joe Long, UNCW
    - Rip currents: Alex Pang, UCSC, & Greg Dusek, NOAA
    - Flooding/CO-OPS stations: Dwayne Porter & Jeremy Cothran, USC, & Greg Dusek, NOAA
    - Situational awareness & object detection: Jeremy Cothran, USC
    - Code implementation: Josh Rhoades, Axiom
- Format: 5 minute presentations from each group
    - Discussion & Q&A on each application

Overall product implementation status

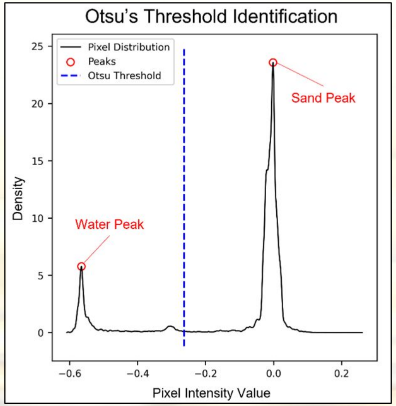
Camera	Shoreline/ wave	Rip currents	Object detection	Flooding	General viewing	Inlet waves
Oak Island east	Y	N	N	N	Y	N
Oak Island west	Y	N	N	N	Y	N
Currituck Hampton inn	Y	Y	M	N	Y	N
Currituck Sailfish	Y	Y	M	N	Y	N
Sheraton Waikiki	M	N	N	N	Y	N
Beaufort Duke Marine lab	N	N	N	Y	Y	N
Charleston Harbor	N	N	Y	Y	Y	N
Holland MI	M	Y	N	N	Y	N
Jennette's North	N	Y	N	N	Y	N
Jennette's South	N	Y	N	N	Y	N
North inlet Winyah bay	N	N	Y	N	Y	N
Point Reyes TMMC	M	N	N	N	Y	N
SC Maritime museum	N	N	N	N	Y	N
UNCW dock North	N	N	M	N	Y	N
UNCW dock south	N	N	M	N	Y	N
Walton Light	M	Y	M	N	Y	N
Cocoa beach	Y	M	N	N	Y	N
Horace Caldwell TX	Y					
Masonboro	N	N	Y	N	Y	Y
Folly 6th	Y	N	Y	N	Y	N
Rosemont Peace 1	N	N	N	Y	N	N
Rosemont Peace 2	N	N	N	Y	N	N
Rosemont Peonie	N	N	N	Y	N	N
LAMC	N	N	N	Y	N	N

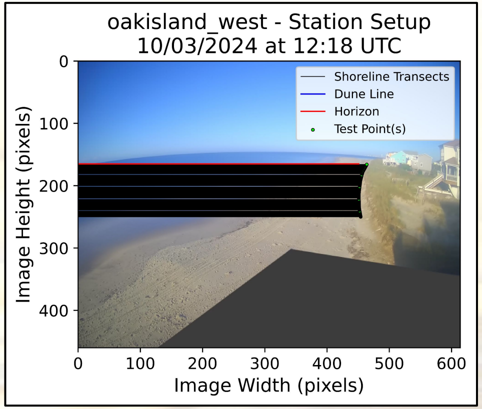
Shoreline change: Joe Long

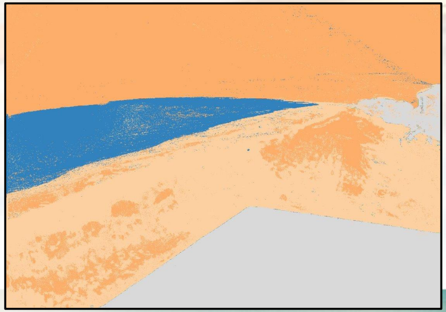
- Otsu's threshold to identify boundary between water and sand on cross-shore transects

- Each camera relies on a station configuration file

- ML methods tested separately (UNCW & UCSC)

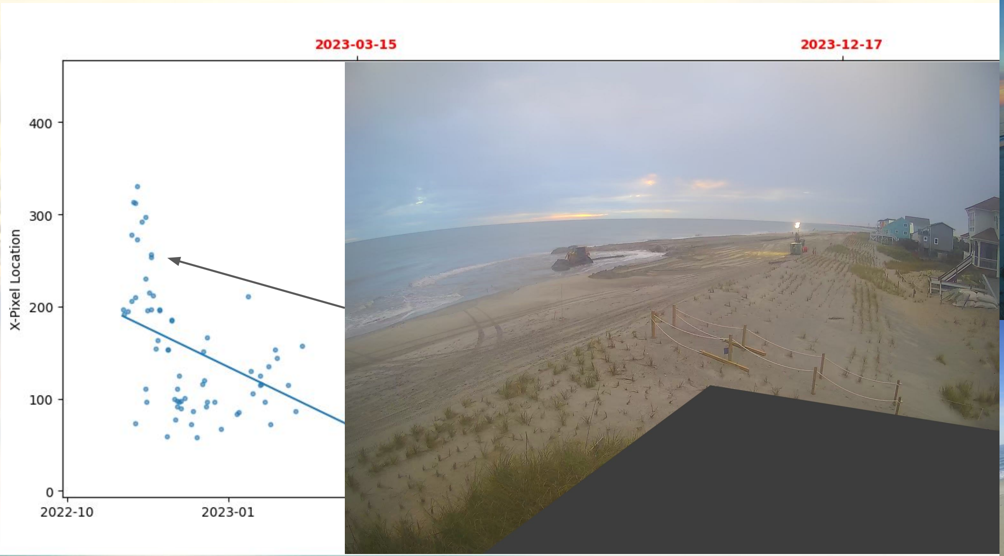






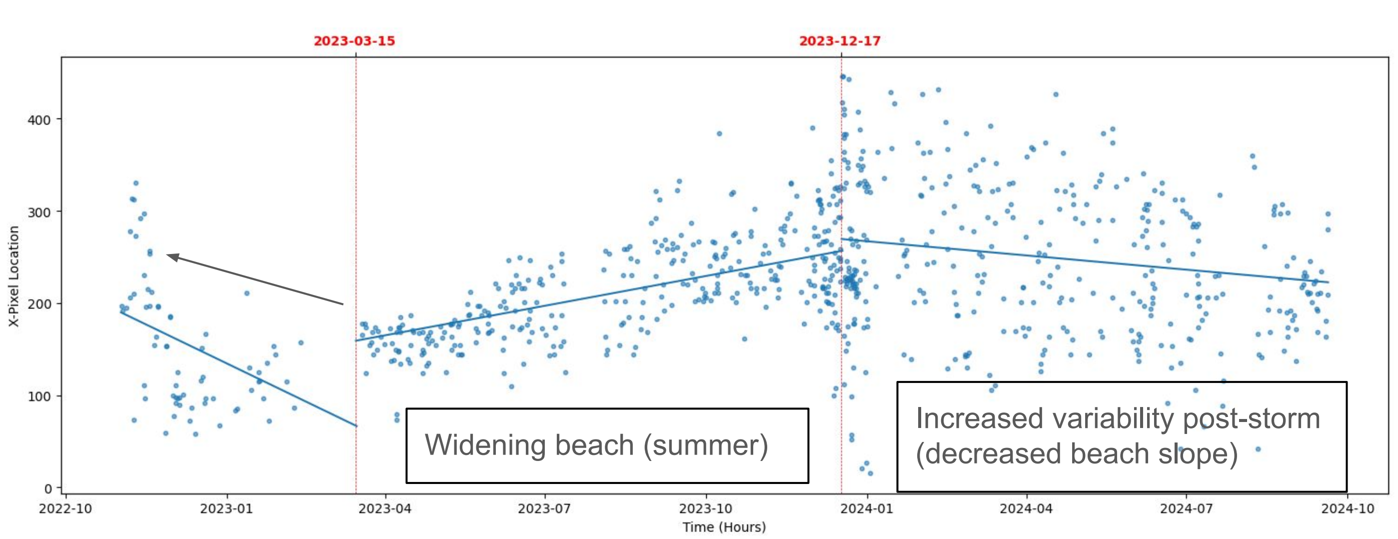
Shoreline change: Trends

- Identify nourishments, seasonal changes, and persistent trends.



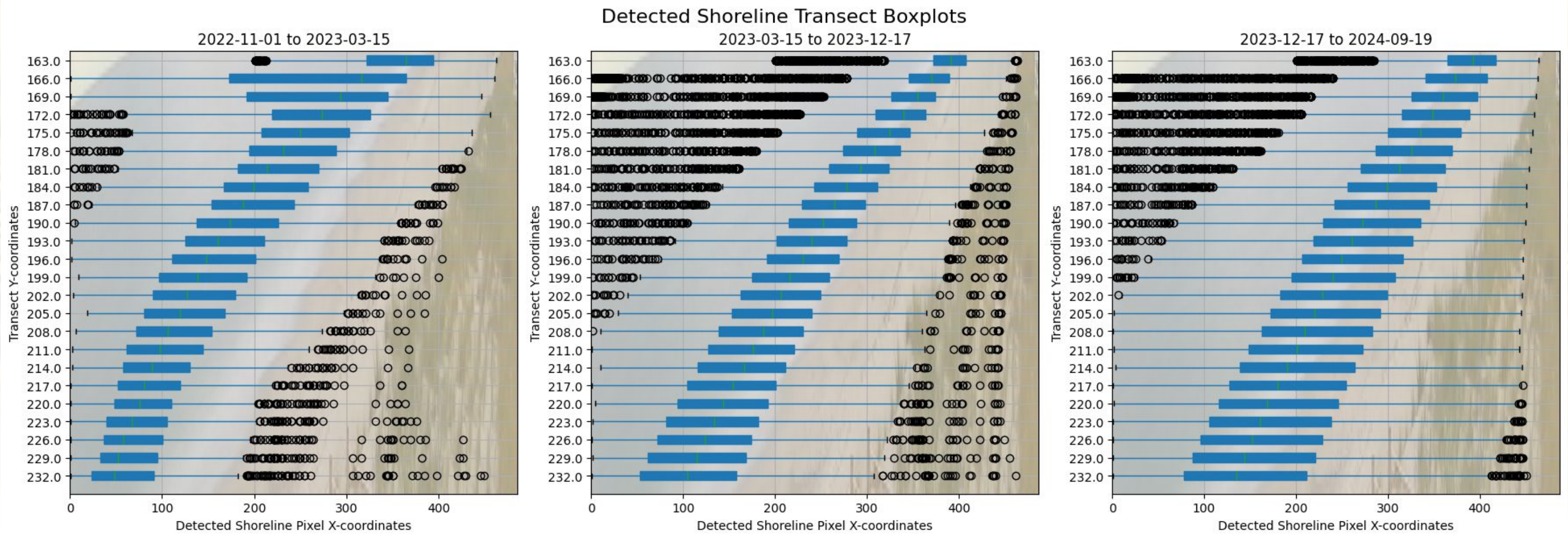
Shoreline change: Trends

- Identify nourishments, seasonal changes, and persistent trends



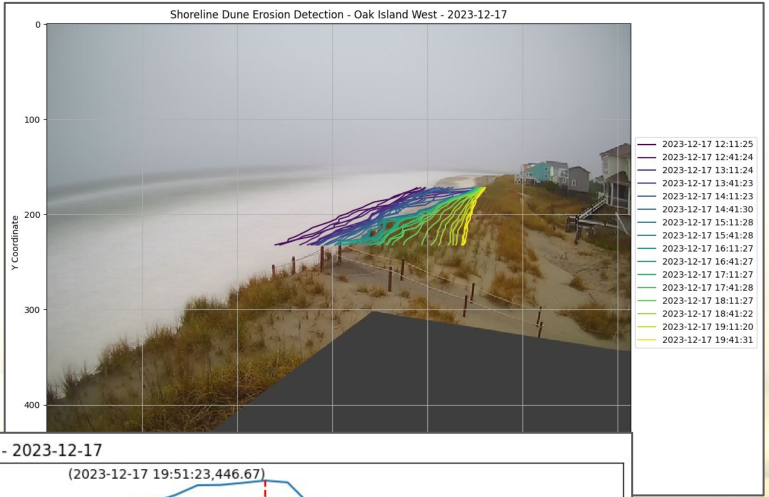
Shoreline change: Trends

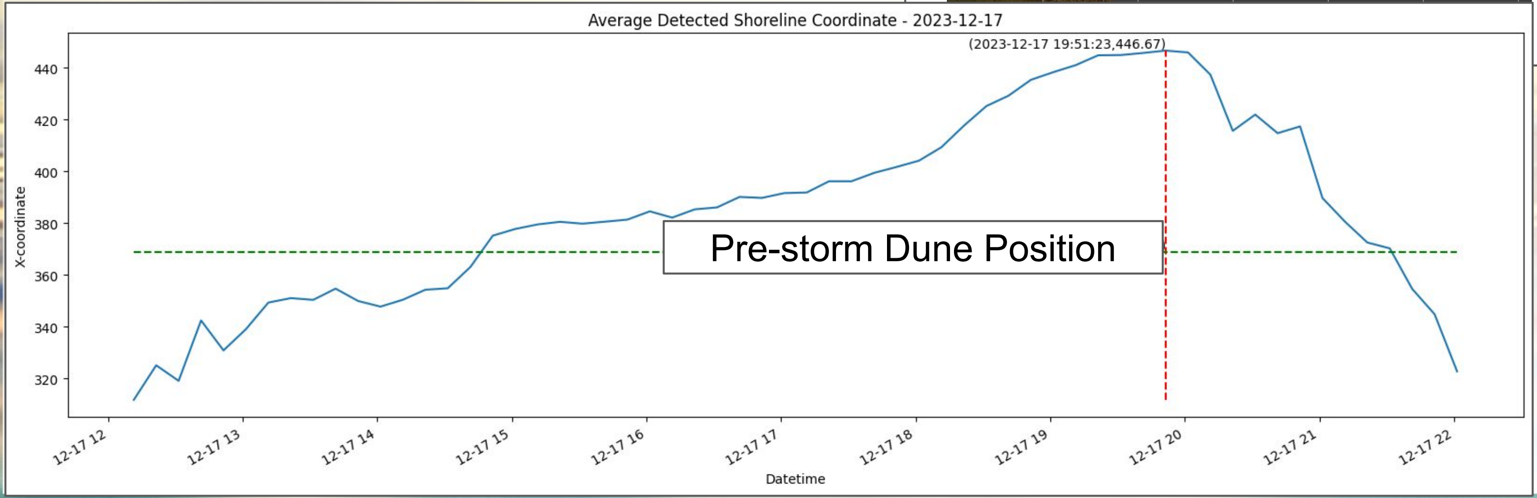
- Identify nourishments, seasonal changes, persistent trends.



Shoreline change: Storms

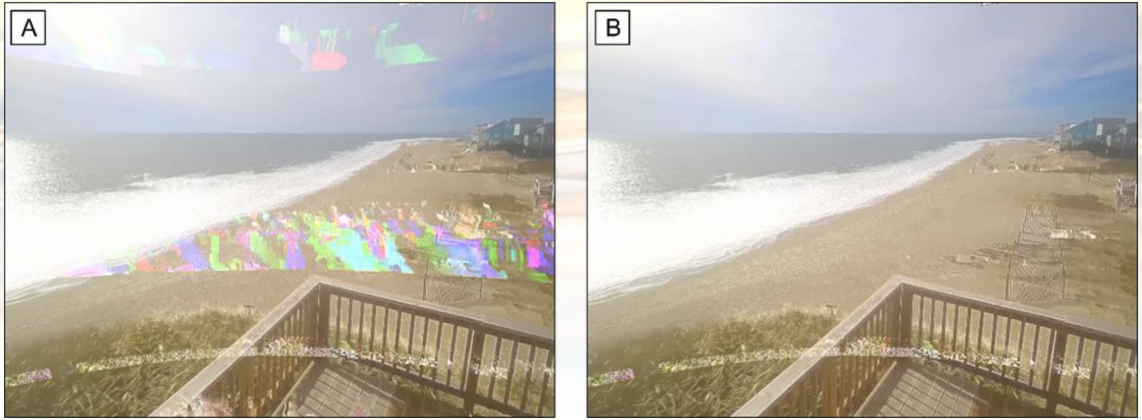
- Visualize timing, duration, magnitude of dune erosion events



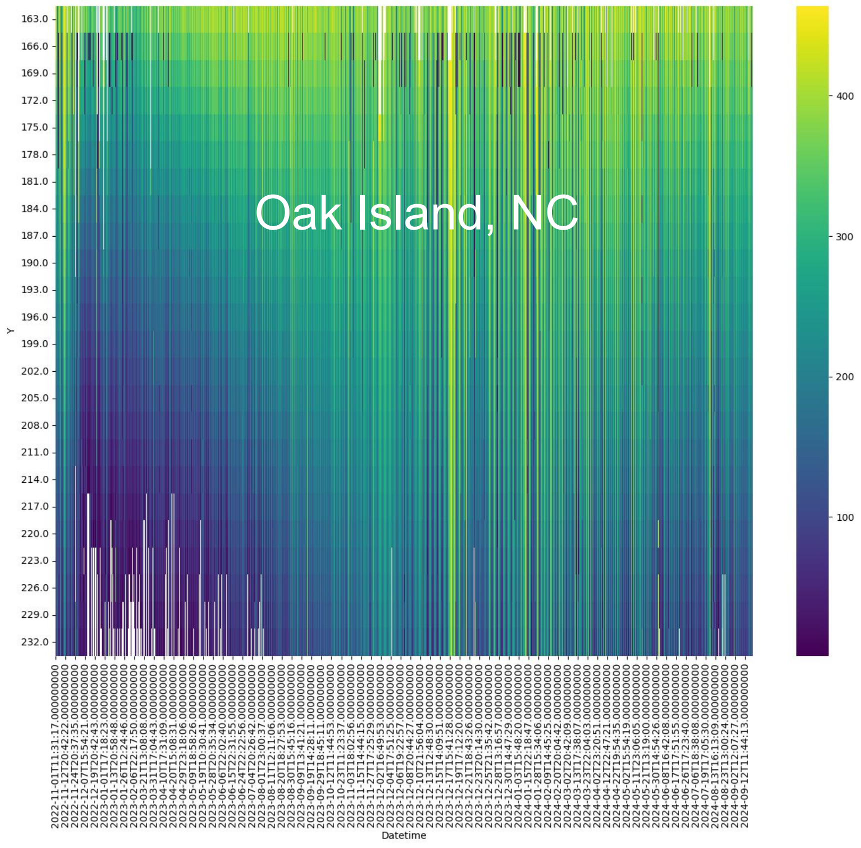


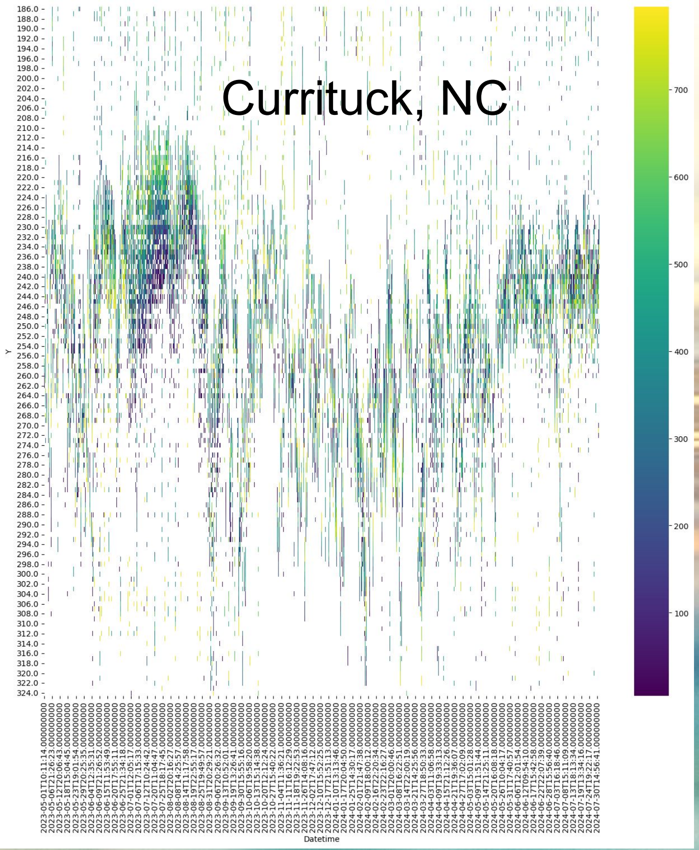
Shoreline change: Limitations

- Pixel-based measurements - not transferable to quantitative change assessments without georectification
- Artifacts of image corruption in time exposure/brightest images
- Rain, fog, camera movement -- the usual suspects!

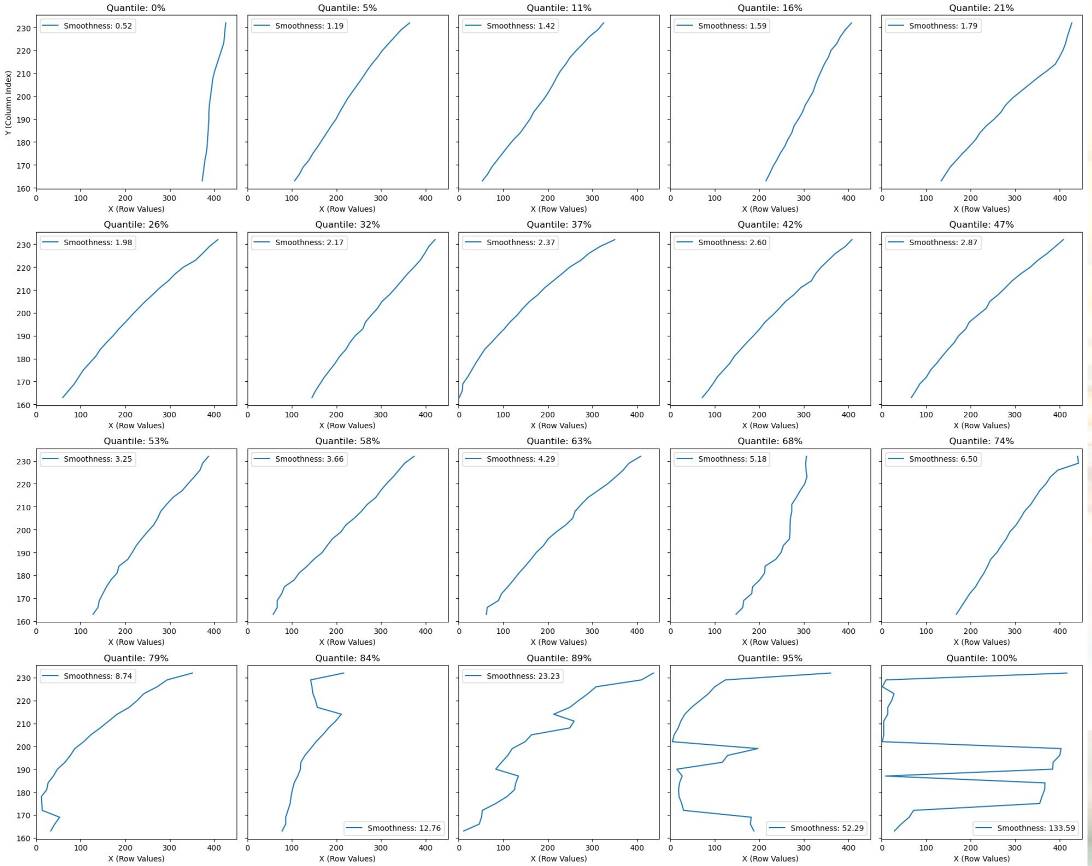


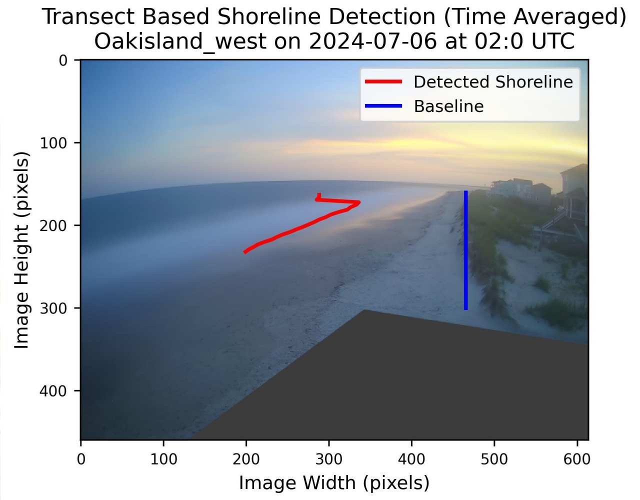
Shoreline change: Not all locations are equal



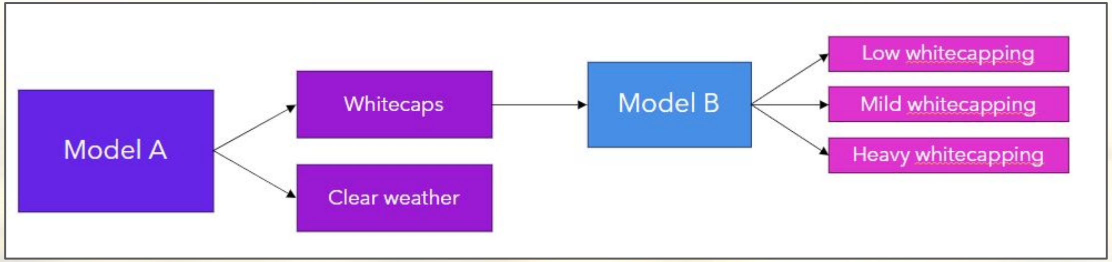


Shoreline change: Good vs. Bad





Inlet Conditions



Rip currents: Alex Pang & Greg Dusek

- Developed 3 classes of methods for rip detection:

Flow Based Approach

Appearance Based Approach

Behavior Based Approach

Rip Detection Status:

- Flow-based requires stable video (not panning, zooming, shaking) and not fully automated
- Behavior-based also requires stable video and GPU (more compute intensive) for near real time performance. Available but not currently deployed.
- Appearance-based can detect rips on a per-frame basis. Currently deployed.

Appearance Based Rip Detection

- Does pretty well when visual signature of rip is strong.
- False negatives when visual signature is weak.
- Requires training data set for each type of rip:
  - Currently bathymetry controlled rips and flash/sediment rips
  - Mostly clear weather. Will need augmentation for environmental factors e.g. fog, cloud/shadows, glare, rain, …
- Currently deployed at: Walton Lighthouse/CA, Currituck County/NC, Holland Beach/MI
- Site specific customization possible/necessary

Flooding

Rosemont, North Charleston
Peace Street camera

Urban flood extent segmentation and evaluation from real-world surveillance camera images using deep convolutional neural network (Yidi Wang,...)
https://www.sciencedirect.com/science/article/pii/S1364815223003250?via%3Dihub

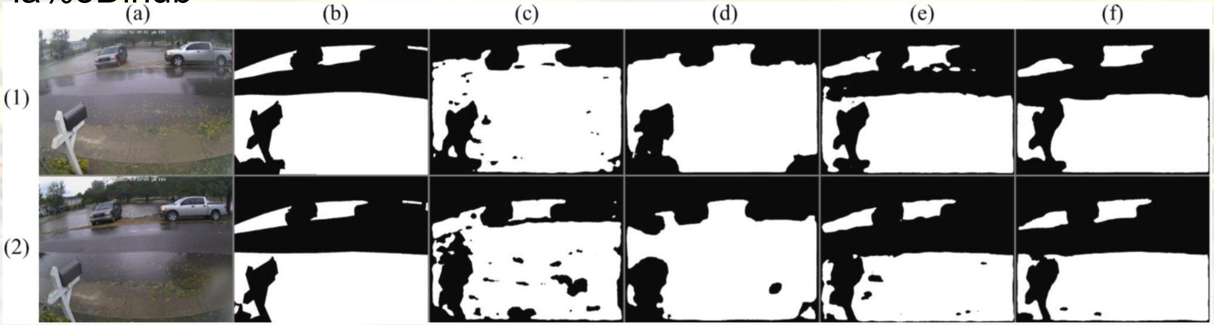
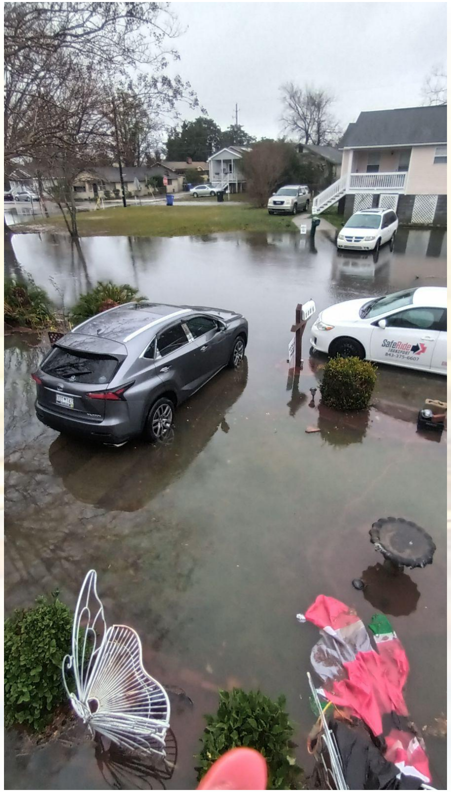


Fig. 5. Examples of model performance in validation. From top to bottom, the four cases are: (1) slightly wavy water surface with normal illumination, (2) still water surface with specular reflection, (3) wet road surface, and (4) night condition. In each case, the six images from left to right are: (a) the original image, (b) ground-truth flood extent, (c) segmented flood extent using Deeplabv3+, (d) image masked with Deeplabv3+ result, (e) segmented flood extent using LinkNet, and (f) image masked with LinkNet result.

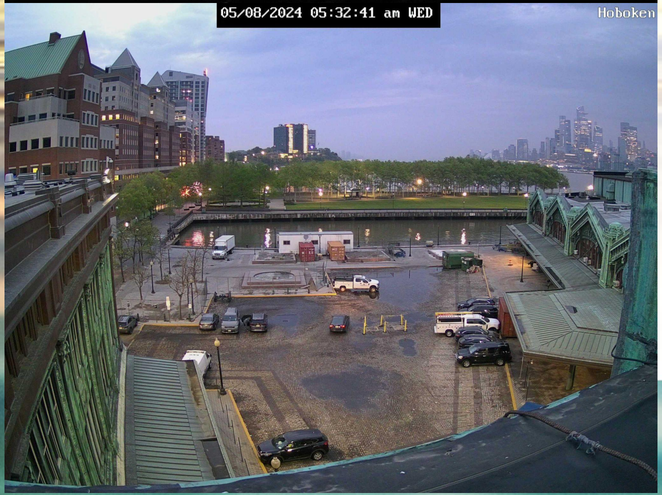
Whaley Way

No flooding, October 2, 2024

flooding, December 17, 2023

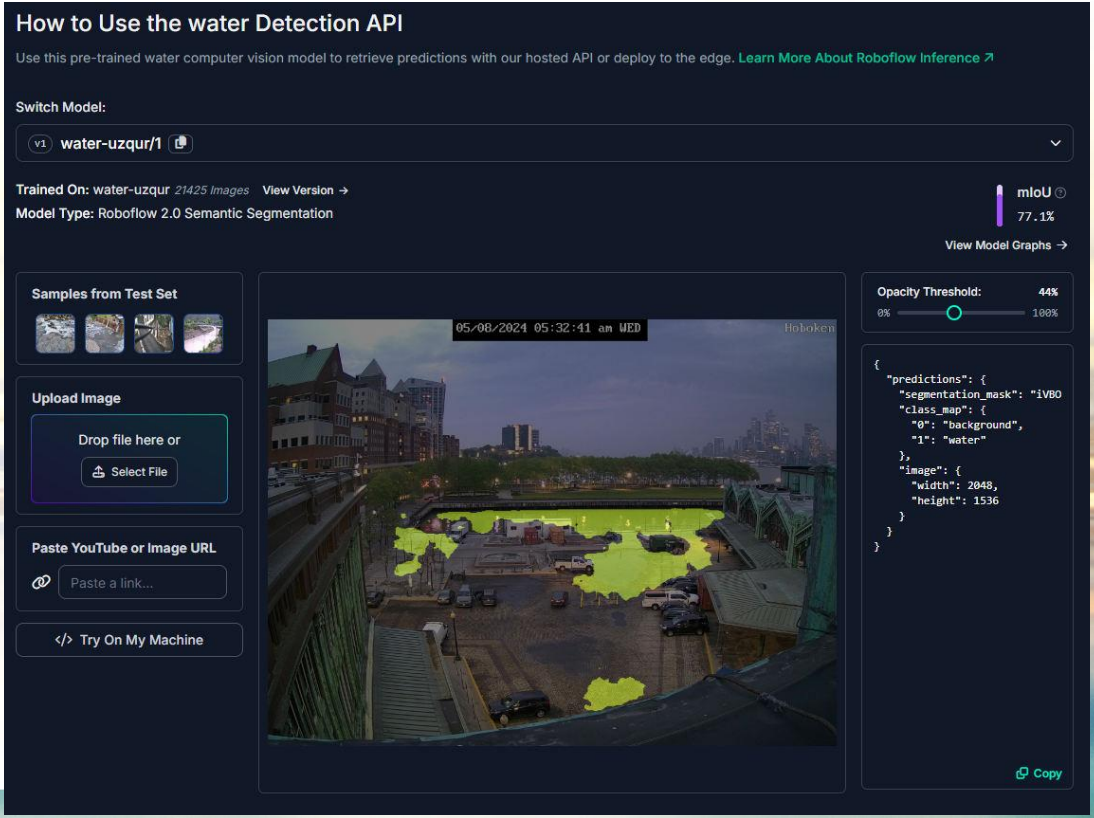


Hoboken, NJ - Stevens Institute of Technology (Philip Orton, Shima Kasaei)
Flooding May 7, 2024, nighttime background lighting sufficient with this camera

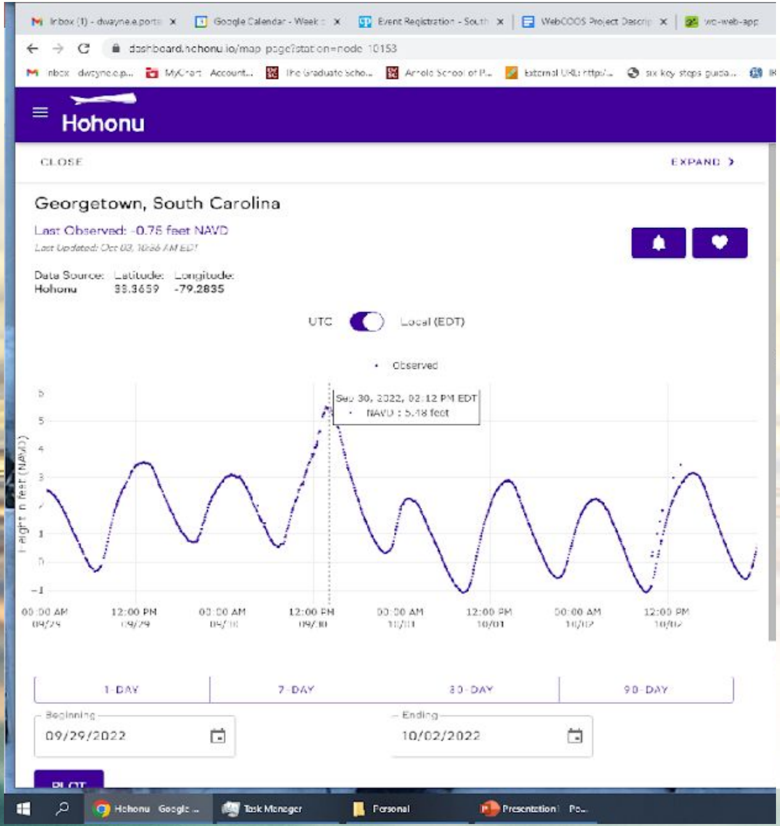


Roboflow https://universe.roboflow.com

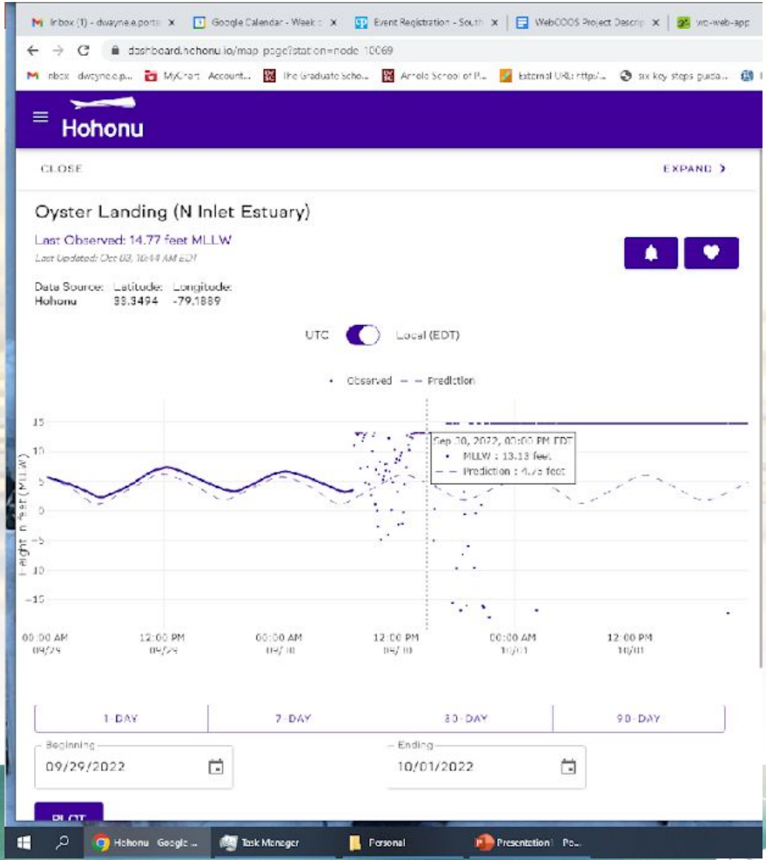
- Publicly available image datasets and models
- Online sample testing and APIs
- Free and paid tier pricing



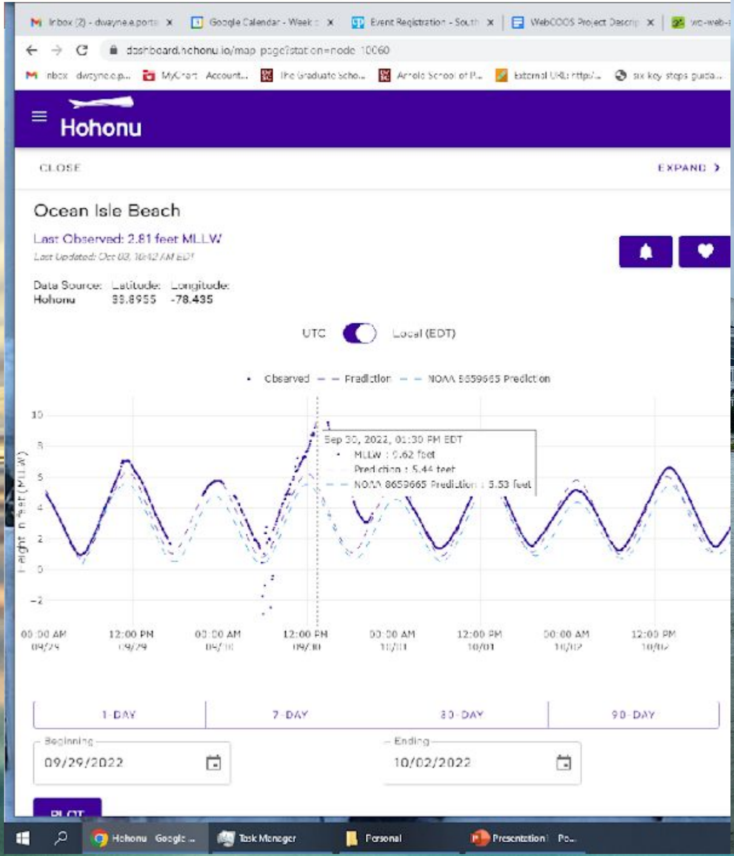
South Carolina Maritime Museum WebCOOS Camera and Water Level Sensor - Hurricane Ian



North Inlet-Winyah Bay NERR – Hurricane Ian



Ocean Isle Beach, NC - Hurricane Ian



Situational awareness/object detection

- YOLO (You Only Look Once)
- https://docs.ultralytics.com/models/yolov8/
- ‘COCO’ dataset (person, vehicle,etc)
- Lowered confidence threshold to 10%
- 10-30% missed or false detections, trends of more use than accuracy

Worked with object detection, though classification and segmentation may be more appropriate depending on the use case

Situational awareness & reporting

UNIVERSITY OF SOUTH CAROLINA

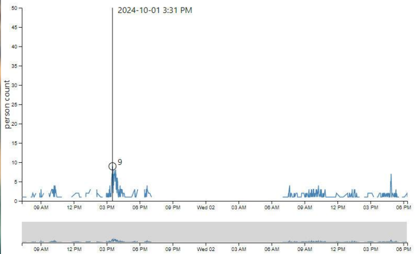
Object Detection (person,...)

Activity graph (time vs counts, links to images)

saludasys.org/counts/index_webcoos.php?cam=noaa_curritu...

Camera: curritück_hampton_inn
Object type: person
Choose other date range(30 day window)
Start Date
End Date
Draw Graph
Graph link

Date Range 2024-10-1 to 2024-10-4
Timezone local to camera



To zoom in/out the graph area, click-hold and drag the gray box edges. To move the window, click-hold and drag the gray box.

The links below are the associated datetime and count for the selected date range. The link will show the image and detections associated with the count in a new browser tab. The image may be slow to load and if the image fails to load, try reloading the tab.

2024-10-01 3:30 PM 7
2024-10-01 3:31 PM 9
2024-10-01 3:33 PM 4
2024-10-01 3:34 PM 2
2024-10-01 3:35 PM 4

Alert/notification thresholds

saludasys.org/notifypage/alert_user_webcoos.php?action=get&session=RI8tbwU62B

Alerts

Type	Location		
person_count_gt_20	noaa_currituck_hampton_inn	Activate	Delete
person_count_gt_20	noaa_currituck_sailfish	Activate	Delete
person_count_gt_30	noaa_currituck_sailfish	Mute	Delete
person_count_gt_5	noaa_currituck_sailfish	Activate	Delete

Add Alert

Please choose alert type and location and click the 'Add' button to add the alert. Maximum of 8 alerts total. In the alert listing above, the 'Mute' button will deactivate the alert and can be toggled back on with the replaced 'Activate' button. The 'Delete' button will delete the alert.

Type | Location | Add

Alert Types

- Person count > 5, the detected count of people is greater than 5, alert resets after 16 hours
- Person count > 20, the detected count of people is greater than 15, alert resets after 3 hours
- Person count > 30, the detected count of people is greater than 20, alert resets after 1 hours

Locations

To review WebCOOS camera locations further
Currituck Sailfish: Activity Graph
Currituck Sailfish: Camera

Currituck Hampton Inn: Activity Graph
Currituck Hampton Inn: Camera

Text/Email notification

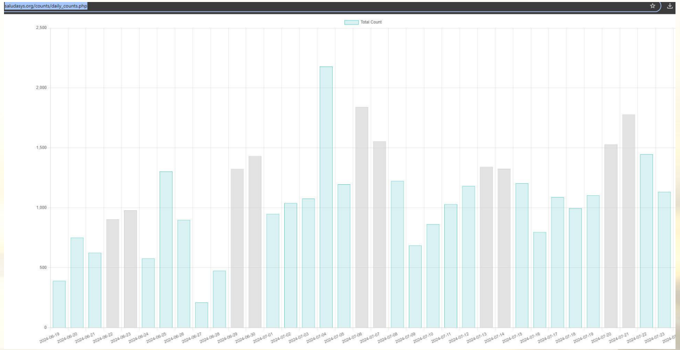
auto-alert: person count > 30 (at 34) at noaa_currituck_sailfish Inbox × midnightdreary7 ×

midnightdreary7@gmail.com Sat, Aug 10, 2:51 PM
to

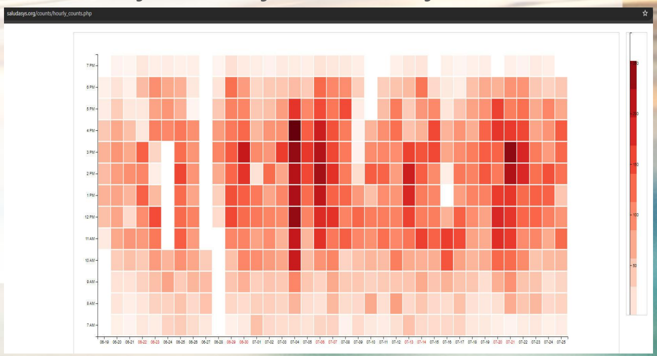
person count > 30 (at 34)
location:noaa_currituck_sailfish
count: https://saludasys.org/counts/detect_webcoos.php?filename=currituck_sailfish-2024-08-10-184543Z.jpg&obj_type=person&station=noaa_currituck_sailfish&action=get
activity: https://saludasys.org/counts/index_webcoos.php?cam=noaa_currituck_sailfish
camera: https://webcoos.org/cameras/currituck_sailfish/

Notification settings: https://saludasys.org/notifypage/login.php?app_type=webcoos

Daily status or activity summary

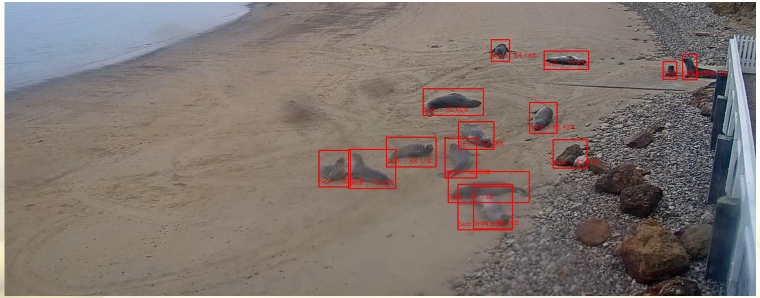


Hourly activity summary



Problem issues

Good detection(seals)



Missed detections due to lack of contrast(lighting) or change in distance or view angle

Missed detections with distance(less pixels/info)

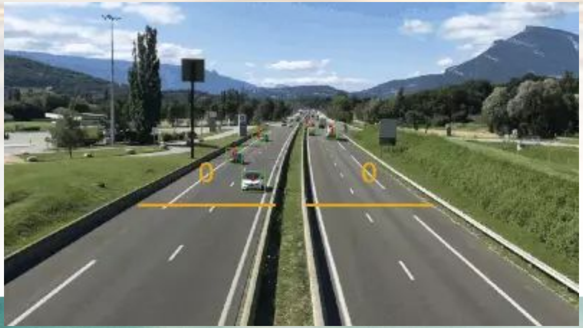
False detections - shadow contrasts or artifacts?

Problem issues

Tried car detection/counting at Isle of Palms beach parking area(lot capacity), but poor results due to

- Low camera angle view
- Occluding vegetation
- Quickly curving road(object rotation)
- Changing lane conditions(available camera view) depending on activity

Optimal traffic view angle - from above to lessen occlusion with minimal object visual rotation/variance



https://github.com/nathanrooy/rpi-urban-mobility-tracker
https://nathanrooy.github.io/posts/2019-02-06/raspberry-pi-deep-learning-traffic-tracker

Tried Amazon rekognition(persons at beach) - object detection on Amazon Web Services cloud, but default results not as favorable as custom approach and ‘custom labels’(on Amazon) more expensive
https://aws.amazon.com/rekognition

Additional approaches

- SAHI(Slicing Aided Hyper Inference) slices an image into a set of smaller overlapping images to help detect small objects in larger images

- Mosaic approach - break image into a zones/mosaic(near,far,...) applying different SAHI image size,confidence threshold,...

- Add/Update to imageset used for training
  - Add object classes to differentiate with visually similar objects (e.g. seal vs rock)
- Try other datasets/weight files

Image segmentation(rip tides,flooding,...)

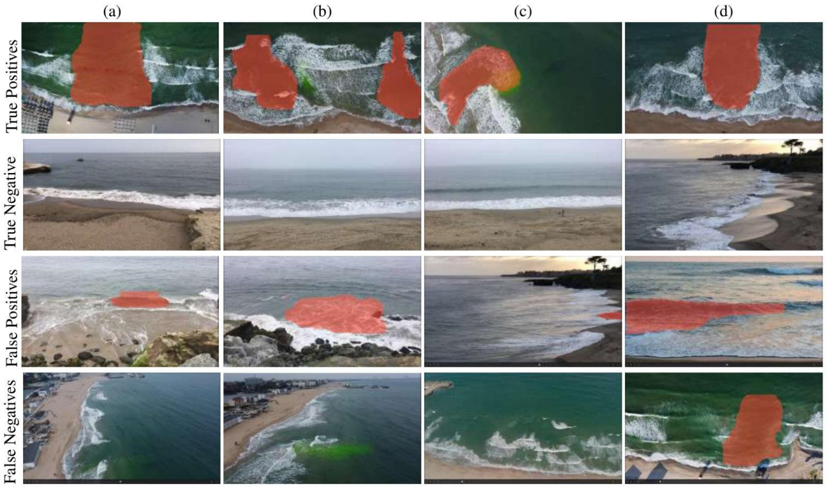


Figure 4. Example of results on testing dataset (using the nano model). On the first two rows we have the correct predictions, the true positives and the true negatives. On the last row we have the incorrect predictions, the false positives and the false negatives. Notice column (d) on the last row, where the model manages to correctly predict one rip current, but misses the second one.

Technology pipeline - could mix and match from different providers

Instrumentation

- Camera resolution
- Perspective / distance
- Programmability/OS/'edge'?
- **Power / connectivity**
- Installation /maintenance

Video/image collection

Images can be pulled at timed intervals from video via FFMPEG

online/near real-time
push - RTMP/FTP,...
pull - RTSP/HTTP,...

offline
SD card/manual for
deployable/trailcam

Storage

- Cloud(Amazon S3 object/file buckets)
- Recent / Archived

Processing(AI/ML,...)

- Cloud / Local, GPU / CPU

Software

- Model (YOLO(You Only Look Once) v8 https://www.ultralytics.com/yolo)
- Training set / Weights File (COCO - Common Objects in Context cocodataset.org)
- https://universe.roboflow.com
- Develop model processing workflow once and reference 'weight' files for matching object/class types

Summary

Database(Postgres) - store/query
location,timestamp,object(s)
count,image bbox pixel x/y coords,
confidence score (summary detection
json output)

Products(may reference any previous layers)

- Graphs/Displays
- Summary(Hour,Day,Month) / Reports
- Alerts/Notification

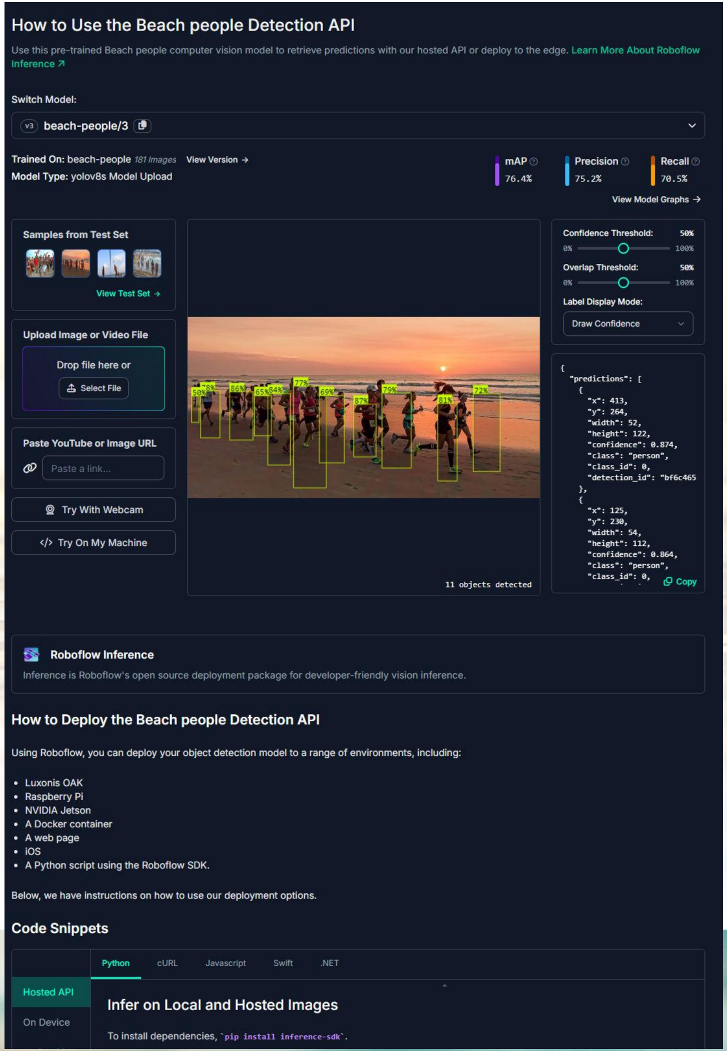
Train/try other datasets/weight files
https://docs.ultralytics.com/modes/train/

Roboflow https://universe.roboflow.com

- Publicly available image datasets and models
- Online sample testing and APIs
- Free and paid tier pricing

Cloud - develop, train, deploy AI/ML applications
https://www.paperspace.com

ChatGPT,Claude,Gemini,...
Inform, aid and suggest development in discussion and code



Code Implementation (challenges and details)

- Challenges with Rip Current Detection
  - Rip current detection requires NVIDIA GPU acceleration (40+ CPU cores cannot serve as a replacement)
  - Have to support multiple rip current models at once
  - Multi-stage products
    - input video → 15-second time average → ML model (general) → detection output
    - input video → once-a-minute still → ML model (walton) → detection output
  - Models, as developed, work with real-time video streams, but Axiom’s pipeline isn’t wired for that
- Challenges with Shoreline
  - **[Filtered]** Multi-stage products:
    - [Input video, longer than 9 minutes, selected from the top of the hour] → ~10 minute time average → shoreline detection algorithm → shoreline detection product
  - GPU provides negligible improvement, and high CPU during time average and brightest pixel calculations.
  - Heavy config lift for each new camera.
- Challenges with General Object Detection
  - Works well, but have to tweak detection in subtle ways:
    - Filtering out erroneous object detections (can’t have trains on beaches, etc.)
    - Had to use a “windowing” feature to better identify objects in frame.
  - PTZ schedules can make for widely different object detection results (as opposed to a fixed camera view)