Developing a Citizen Science Web Portal for Manual and Automated Ecological Image Detection

Marshall Mattingly III[*], Andrew Barnas[†], Susan Ellis-Felege[†], Robert Newman[†], David Iles[‡], Travis Desell[*]

Department of Computer Science[*], Department of Biology[†]
University of North Dakota

Department of Wildland Resources and the Ecology Center[‡]
Utah State University

Wildlife@Home

http://csgrid.org/csg/wildlife/

UND UNIVERSITY OF NORTH DAKOTA

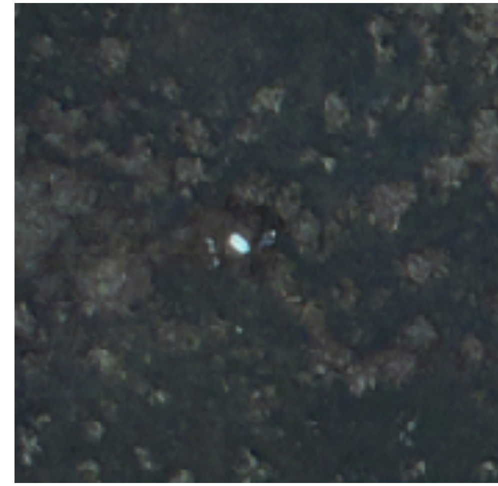
What is Wildlife@Home

- A *citizen science* project that combines both crowd sourcing and volunteer computing
- Users volunteer their brain power by observing videos and images and reporting observations
- Users volunteer their computing power by downloading videos and performing computer vision computations
- A scientific web portal to robustly analyze and compare results from users, experts, and the computer vision techniques

Images collected for research

- All imagery used for this research is from the Hudson Bay area of Manitoba, Canada
- Trail cameras deployed to learn about predators destroying nests
  - Common eider and lesser snow geese
  - 85 cameras
  - 100 nests
  - Primary issues:
    - Cryptic coloration (camouflage), obscuring vegetation
- Unmanned Aerial System (UAS) imagery flown along predetermined transects
  - Lesser snow geese
  - Vegetation and other landmarks
  - **Focus of this research**
  - Primary issues:
    - Cryptic coloration of the blue phase lesser snow geese
    - Small objects in comparison to the images





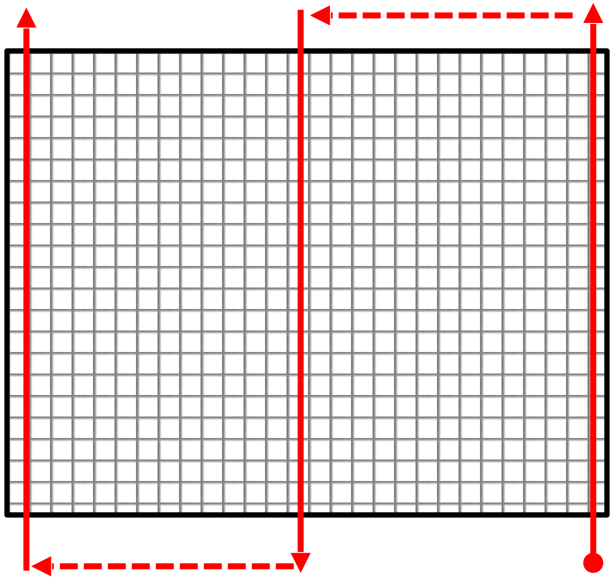
How many Lesser Snow Geese are in these images? **2 in each!**



Changing lighting conditions add to the difficulty of image processing.

UAS Image Collection

- In summer 2015, a Trimble UX5 fixed wing UAS was flown at Wapusk National Park in Manitoba, Canada
- Flights were flown at 75m, 100m, and 125m on pre-defined transects with 80% overlap
- Images were taken with a 16 megapixel Sony red, green, blue camera in the Nadir position



UNIVERSITY OF NORTH DAKOTA

UAS Dataset

- 60,000 images were produced from the flights
- 10 mosaics were created using Trimble
- Over 1 Terabyte of image data, with more data being generated each year
- Too much data for experts to analyze alone!
- That's where *citizen scientist* come in

UNIVERSITY OF NORTH DAKOTA

Creating a UI for Image Observations

- Web-based user interface with touch capabilities for tablets and desktops
- Present the same image to three (or more) users
- Match user observations of a single object
- Extract observed objects from images

Challenges with creating the UI

- Images can be significantly larger than the typical viewport of a desktop monitor, e.g. 1920x1080 pixels
- Observed objects that are too small (only a few pixels square) do not make good candidates for computer vision techniques
- The interface must be usable across a variety of viewports, operating systems, and input devices
- The UI must be usable and extensible for multiple projects and image sources
  - Specifically, the trail cams and UAS imagery

UND UNIVERSITY OF NORTH DAKOTA

Overcoming large images

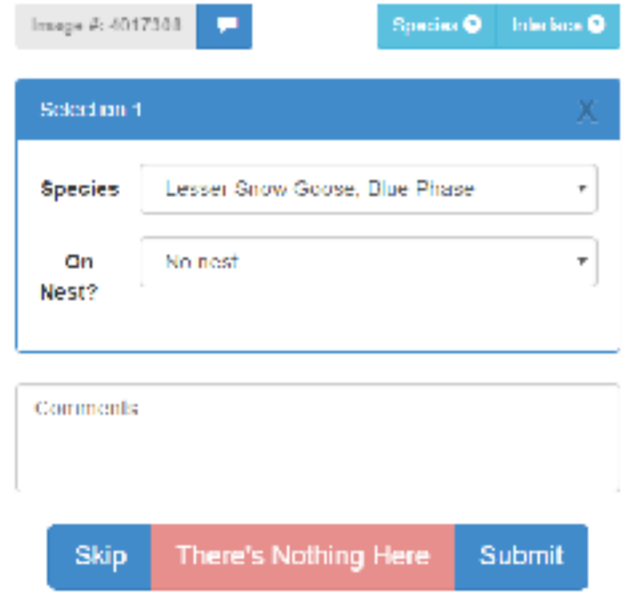
- Large images are split into either 25 or 100 smaller images, depending on the size
- Resultant images are constrained to approximately 1280 pixels
  - The splitting of images is being re-written to force this maximum constraint
- Use an HTML5 Canvas to allow the user to scroll in any direction and zoom the image, in the case that the image is still larger than the viewport
  - This is especially useful on tablets

UND UNIVERSITY OF NORTH DAKOTA

- HTML5 Canvas Element
  - zoomable (scroll-wheel or pinch)
  - pannable (click-drag or touch-drag)
- Zoom level
  - current magnification level
- Scroll location (X and Y)
  - shows the current location and size of the image shown
  - relative to the true size of the image

Enforcing a minimum size limitation

- Bounding boxes are created by double-tapping the image to signify an object observation
  - User then identifies the species
  - Resizable and movable
- Objects that are too small do not contain enough data to provide good computer vision training
- Observations are therefore limited to a minimum 5-pixel with and height (25-pixels square)



Observation classification

Corresponding bounding box in the interface

Usability on a variety of hardware

- HTML5 and JavaScript are the only languages required to run the UI
  - Modern browsers, including phone and tablet browsers, are compatible with both technologies
  - Usable on Android, iOS, Windows, Linux, Mac, etc.
- Hammer.JS is used to provide touch-capable inputs to the HTML5 Canvas element

Ensuring good observations

- Computer vision techniques require the positive samples (observations) to have a low background-to-object ratio
  - If there is too much background information, the object may be incorrectly identified in the background of images
- Different users provide different bounding boxes with varying degrees of background information for the same objects

Mapping observations

- Use multiple observations to determine a "true" observation
  - Trust multiple users, not just a single user
- Match user observations of the same object
  - Only accept objects which have observations from multiple users
  - Two (2) algorithms tested
- Determine the "true" bounding box for the matched objects
  - Two (2) algorithms tested with multiple parameters

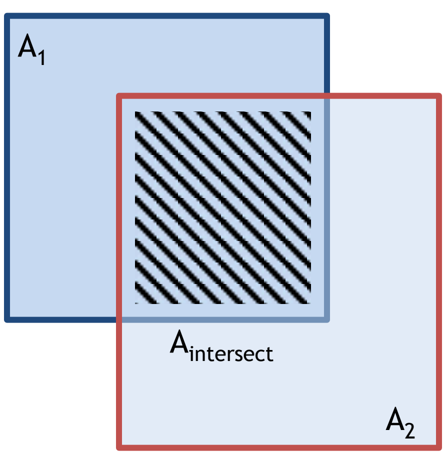
Matching algorithms

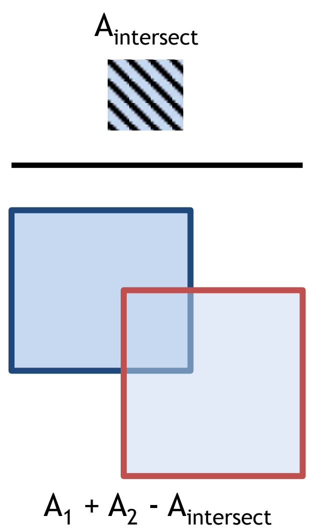
1. Area Overlap
   - Compares the total amount of area of the overlap between two observations
   - Returns the overlapping area as a percentage relative to the total area of the observations
2. Corner-point Distance
   - Calculates the maximum distance between the four corners of two observations
   - Returns the maximum distance calculated

Matching algorithms, cont.

Area Overlap

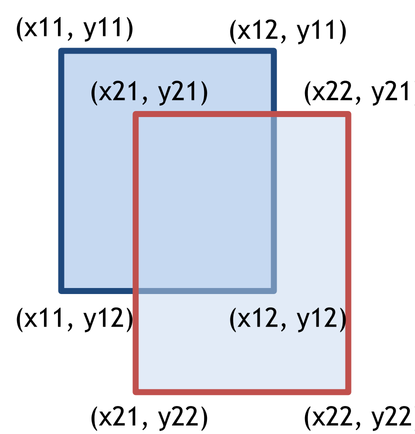
$$
\begin{aligned}
l_{intersect} &= \min\|x_{12}, x_{22}\| - \min\|x_{11}, x_{21}\| \\
h_{intersect} &= \min\|y_{12}, y_{22}\| - \min\|y_{11}, y_{21}\| \\
A_{intersect} &= \max\|0, l_{intersect} * h_{intersect}\| \\
A_{union} &= A_1 + A_2 - A_{intersect} \\
A_{overlap} &= \frac{A_{intersect}}{A_{union}}
\end{aligned}
$$

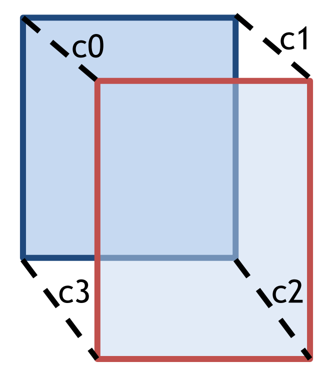




Corner-Point

$$
\begin{aligned}
c_0 &= \sqrt{(x_{11} - x_{21})^2 + (y_{11} + y_{21})^2} \\
c_1 &= \sqrt{(x_{12} - x_{22})^2 + (y_{11} + y_{21})^2} \\
c_2 &= \sqrt{(x_{12} - x_{22})^2 + (y_{12} + y_{22})^2} \\
c_3 &= \sqrt{(x_{11} - x_{21})^2 + (y_{12} + y_{22})^2} \\
c_{max} &= \max\|c_0, c_1, c_2, c_3\|
\end{aligned}
$$





UNIVERSITY OF NORTH DAKOTA

Matching algorithm results

Algorithm	Matches	Non-Matched	False Positives	False Negatives
Actual	400	91	0	0
Area (50%)	352	122	0	54
Area (60%)	329	159	0	92
Area (70%)	266	282	0	214
Area (80%)	186	440	0	370
Area (90%)	81	649	0	566
Point (5px)	238	341	0	272
Point (10px)	379	106	0	24
Point (15px)	404	91	8	0
Point (20px)	414	91	18	0

MATCHES OF THE 811 OBSERVATIONS FROM 142 IMAGES IN THE TEST DATASET

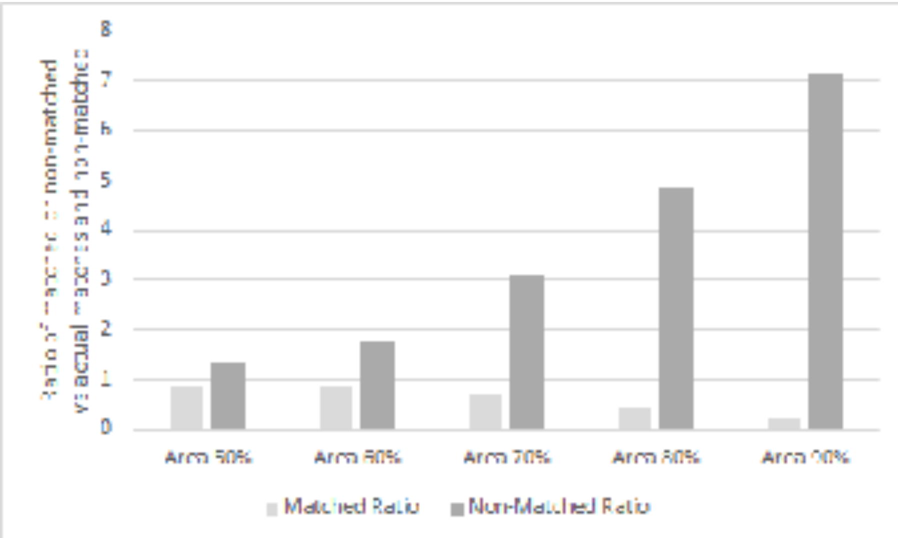
- **Matches** is the number of matched observation pairs
- **Non-Matched** is the number observations without a matched pair
- **False positives** are matches that are not actual matches
- **False negatives** are when the algorithm fails to match observations that should match

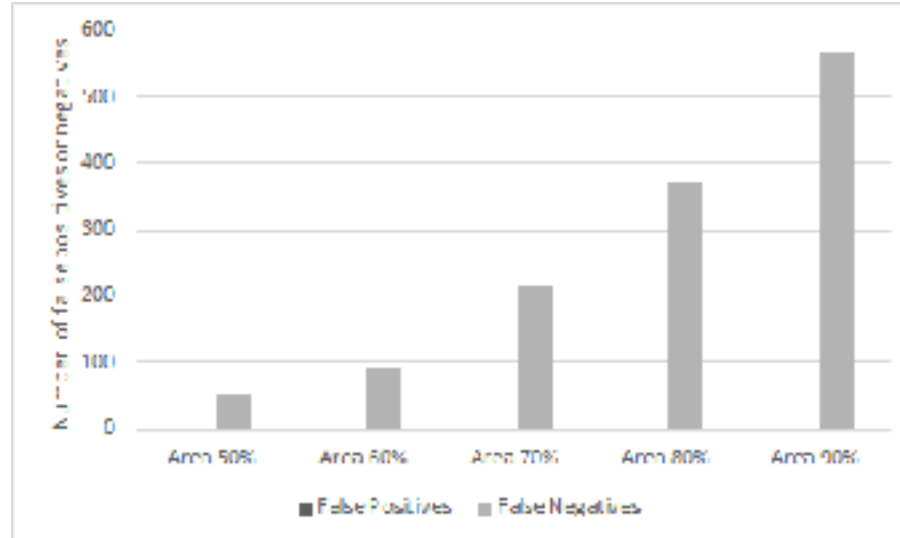
- **Point (10px)** is chosen as the matching algorithm because:
  - provides the highest matched ratio (0.95)
  - provides a low non-matched ratio (1.16)
  - has no false positives and few false negatives

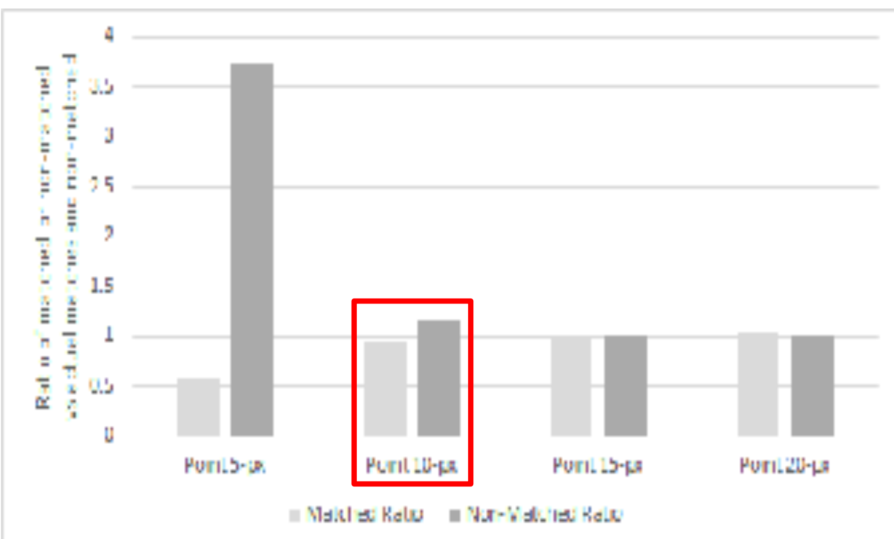
UND UNIVERSITY OF NORTH DAKOTA

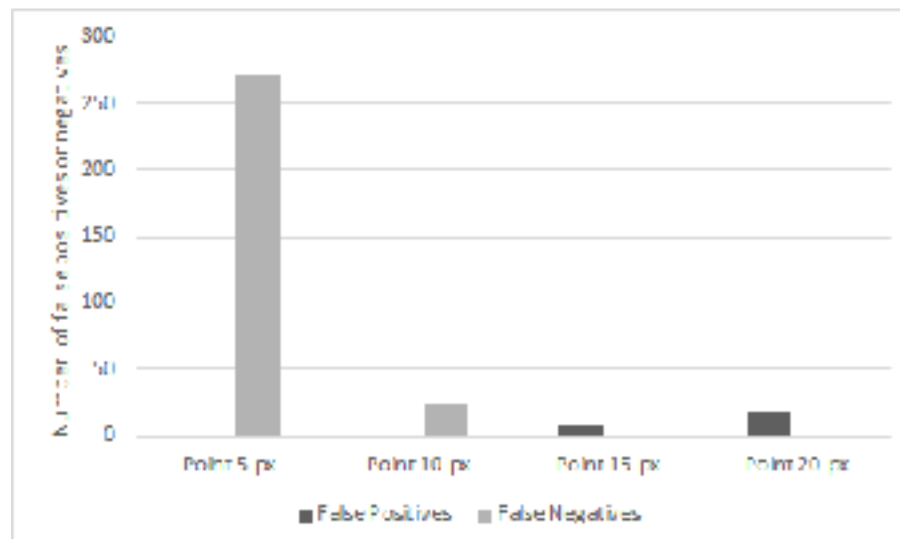
Matches vs Non-Matches

False Positives and Negatives









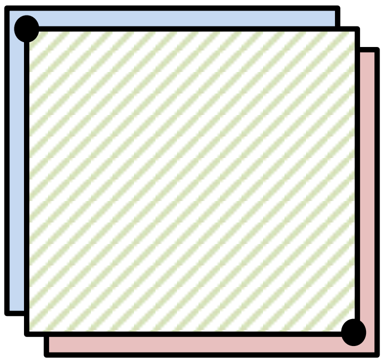
Observation extraction

- Now that we have a set of matched observation, we have to use the aggregate bounds to create the “true” bounds

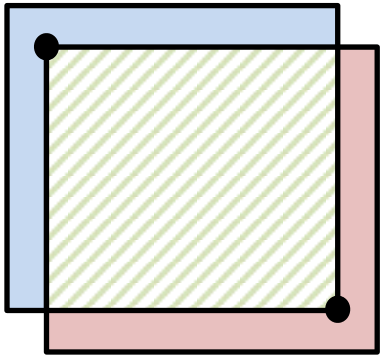
1. Average extraction method
   - Averages the location of each corner
   - EASY TO SKEW WITH **TOO MUCH BACKGROUND**
     - All inputs have the same weight
     - Relies on all users to give relatively good input
2. Intersection extraction method
   - Pulls out the intersection of each observation
   - EASY TO SKEW WITH **TOO LITTLE POSITIVE DATA**
     - Relies on a single user having good input
     - Minimizes background noise
     - A single box too small can give less positive data than is present

Observation extraction, cont.

Average



Intersection



Observation extraction results

- Difficult to analyze the amount of negative space programmatically
- Initial manual inspection shows that the **intersection method** is significantly better than the average method

Conclusions

- Citizen scientists do a good job finding objects
  - only 11.2% of observations failed to be matched with an observation from another user
- However, there is relatively high variability in the bounding boxes around the objects
  - Fatigue, human error, speed, lack of training, etc.
- Using the Corner-Point matching algorithm with a 10-pixel parameter and the Intersection extraction method provide the best positive data set

Future Work

- Compare the results of citizen scientist with those of trained experts
  - show that citizen scientists produce observations similar to the experts
- Train a neural network using the objects extracted from the citizen scientist observations
  - initial work has begun using OpenCV
  - more citizen scientist observations required to build the positive dataset

Acknowledgements

Funding was provided by North Dakota EPSCoR, the Hudson Bay Project, Central and Mississippi Flyways, National Geographic, and UND's College of Arts and Sciences.

Wildlife@Home is currently being supported by NSF award no. 1319700 through the Division of Intelligent Information Systems's Information Integration and Informatics (III) program.

We appreciate the support and dedication of the Wildlife@Home citizen scientists who have spent significant amounts of time reviewing images and classifying objects within them.

UND UNIVERSITY OF NORTH DAKOTA

QUESTIONS?