AIP Project : Computer Vision for Underwater Top Plan Automation

By Daniel Milton

The goal of this project was to create a program to scan underwater digital terrain of archaeology sites and return a map of all significant objects, also known as a top plan

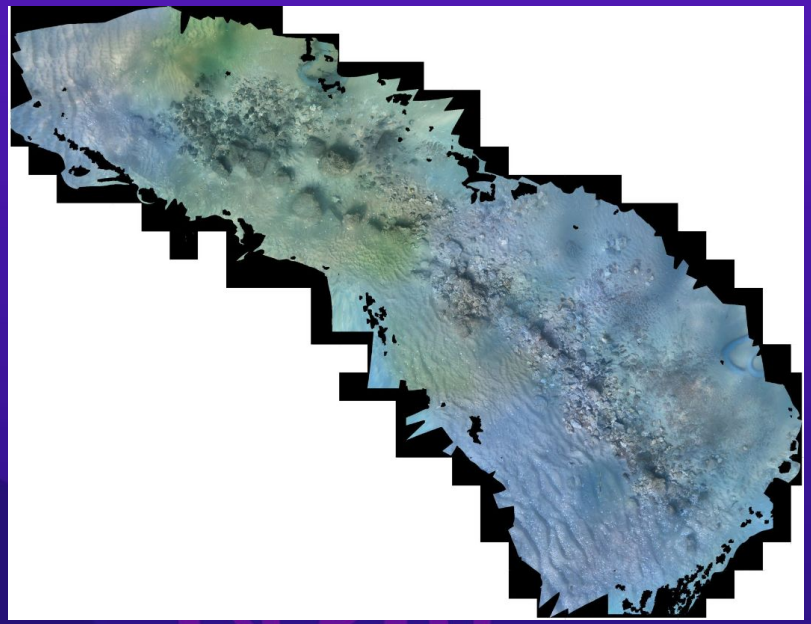
Introduction

- Archeologists used 3d Photogrammetry technology to stitch together GoPro footage of underwater sites
- Used this to create a high definition 3d terrain model
- An archeologist would look at this model and circle all significant points to be referenced to later (Top Plan)

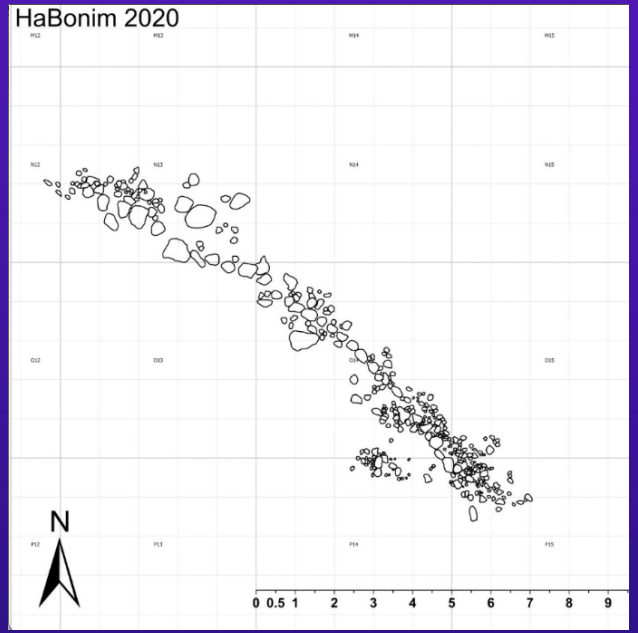
Recording terrain with GoPros

Project Plan

- Automate the top plan using software



Orthophoto



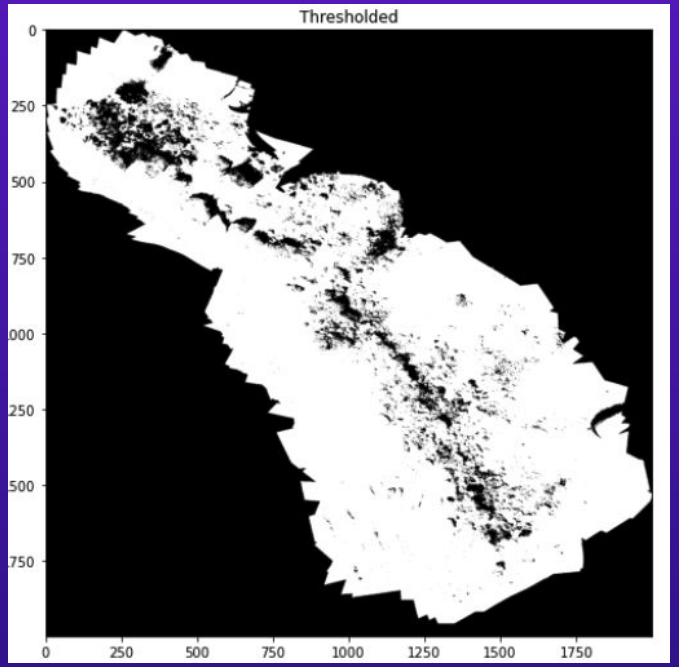
Top Plan

My Approach

- Used famous computer vision Python library called OpenCV
- 3 strategies that stood out to me

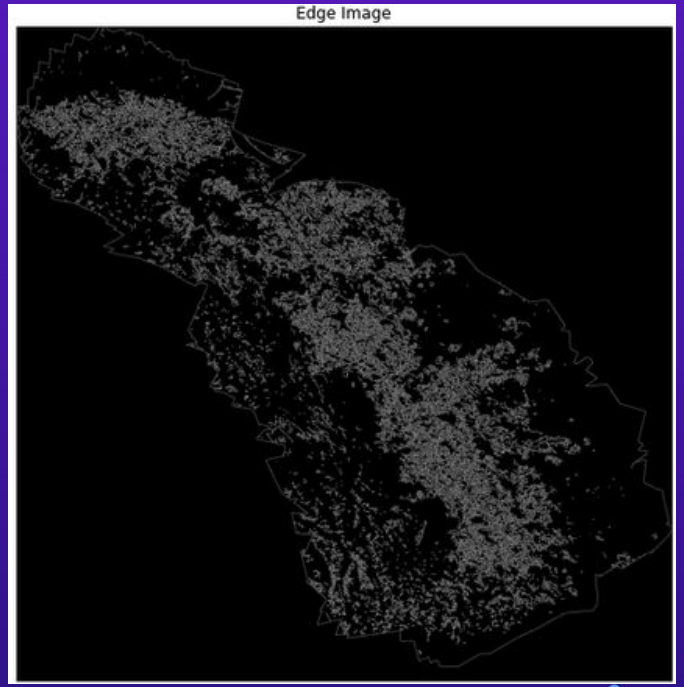
Image Thresholding

- Can change a colorful image to a binary image
- Uses a threshold value
- If above the value, pixel is set to 1
- If below the value, pixel is set to 0



Edge Detection

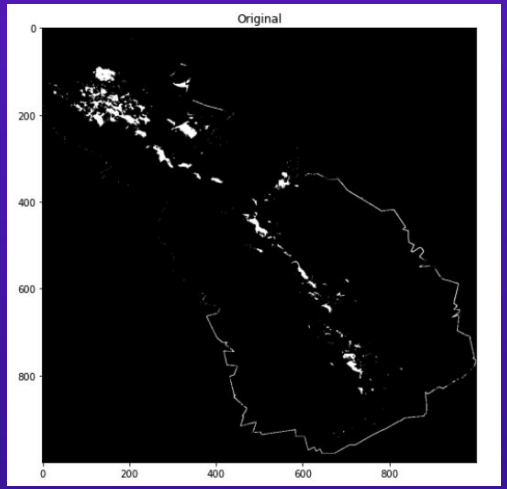
- Applied a gaussian blur beforehand to reduce noise
- Made many tweaks
- Results did not turn out good



Combination of several techniques

1st

Applied a Gaussian blur



2nd

Converted the image from RGB color scale to HSV

3rd

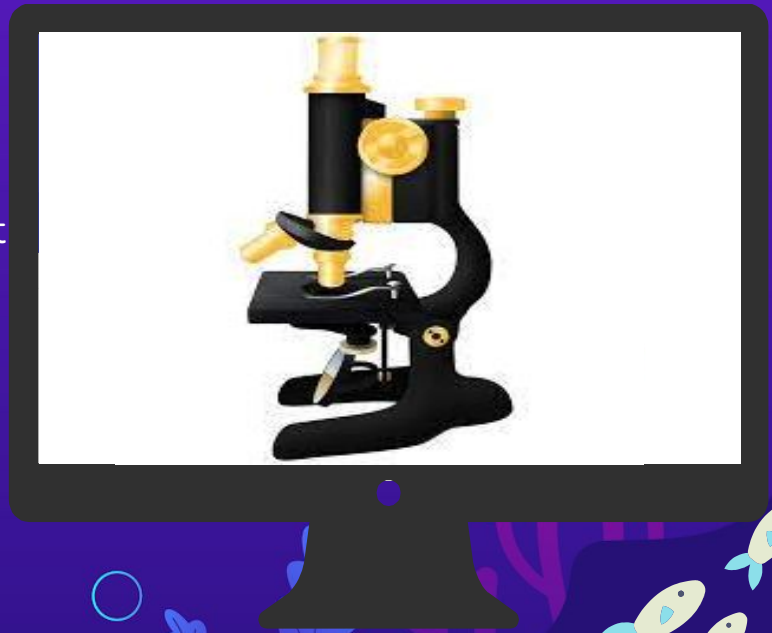
Thresholded the image

4th

Then Inverse the Image

ImageJ

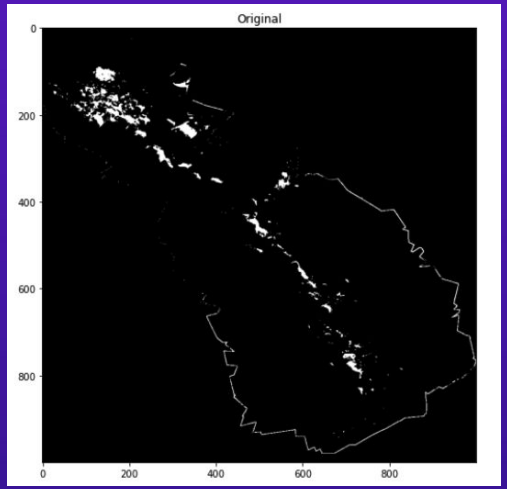
- Let me manipulate images without having to hardcode every change
- I found the perfect values to give me the best result



Combination of several techniques

1st

Applied a Gaussian blur



2nd

Converted the image from RGB color scale to HSV

3rd

Thresholded the image

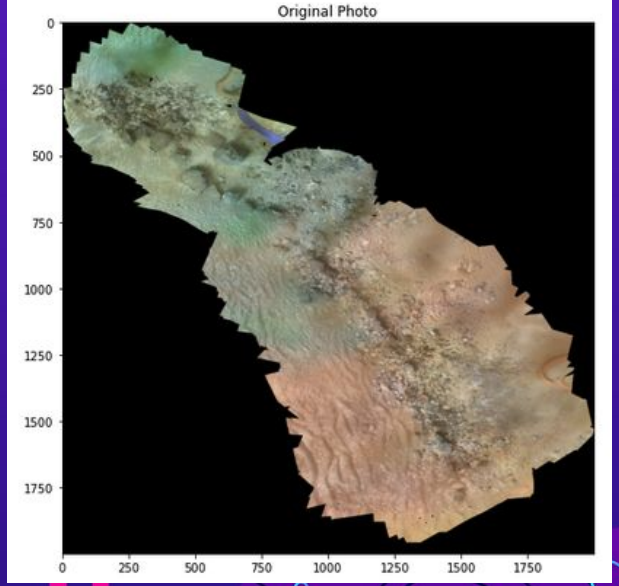
4th

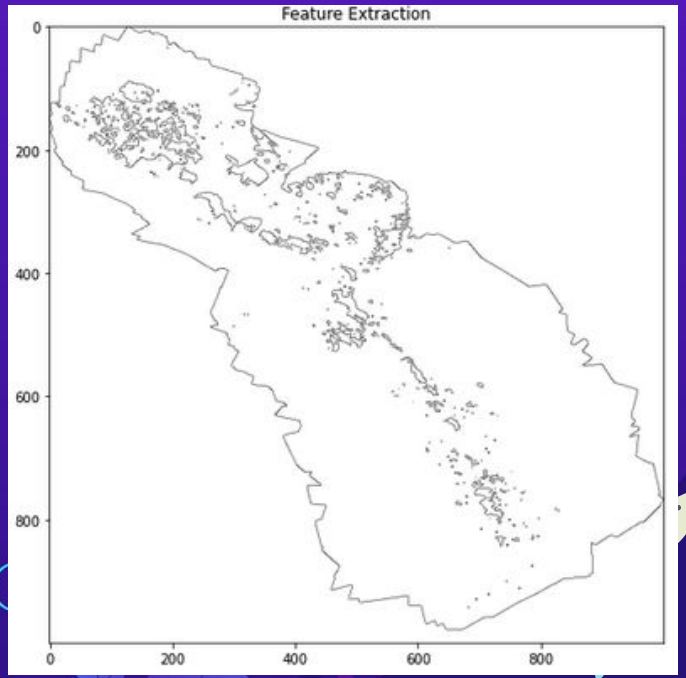
Then Inverse the Image

Results

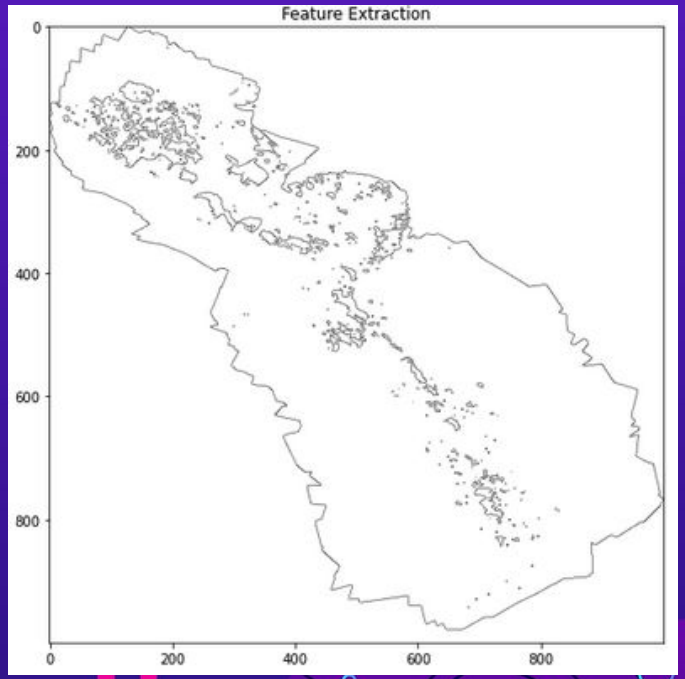
```
1 topPlanAutomizer("ortho1.tif")
```

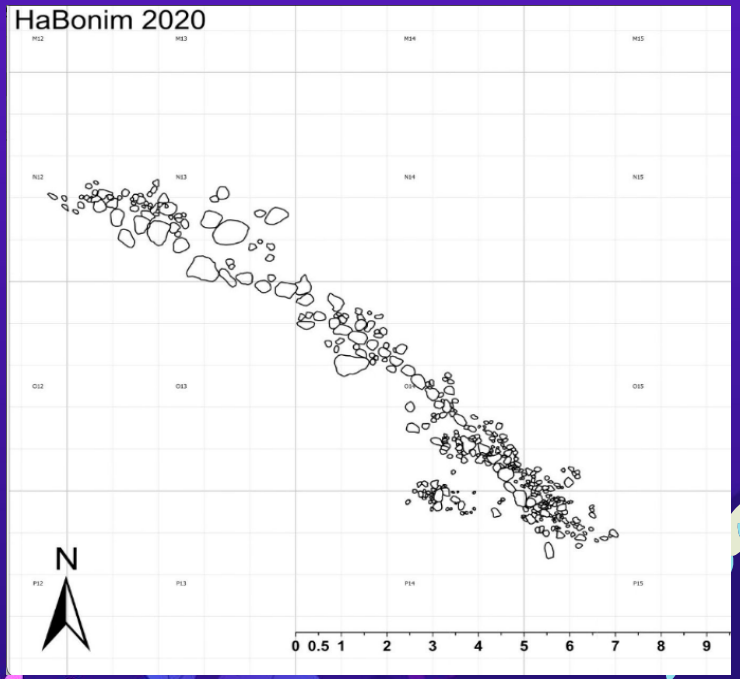
```
'Feature Extraction Successful'
```





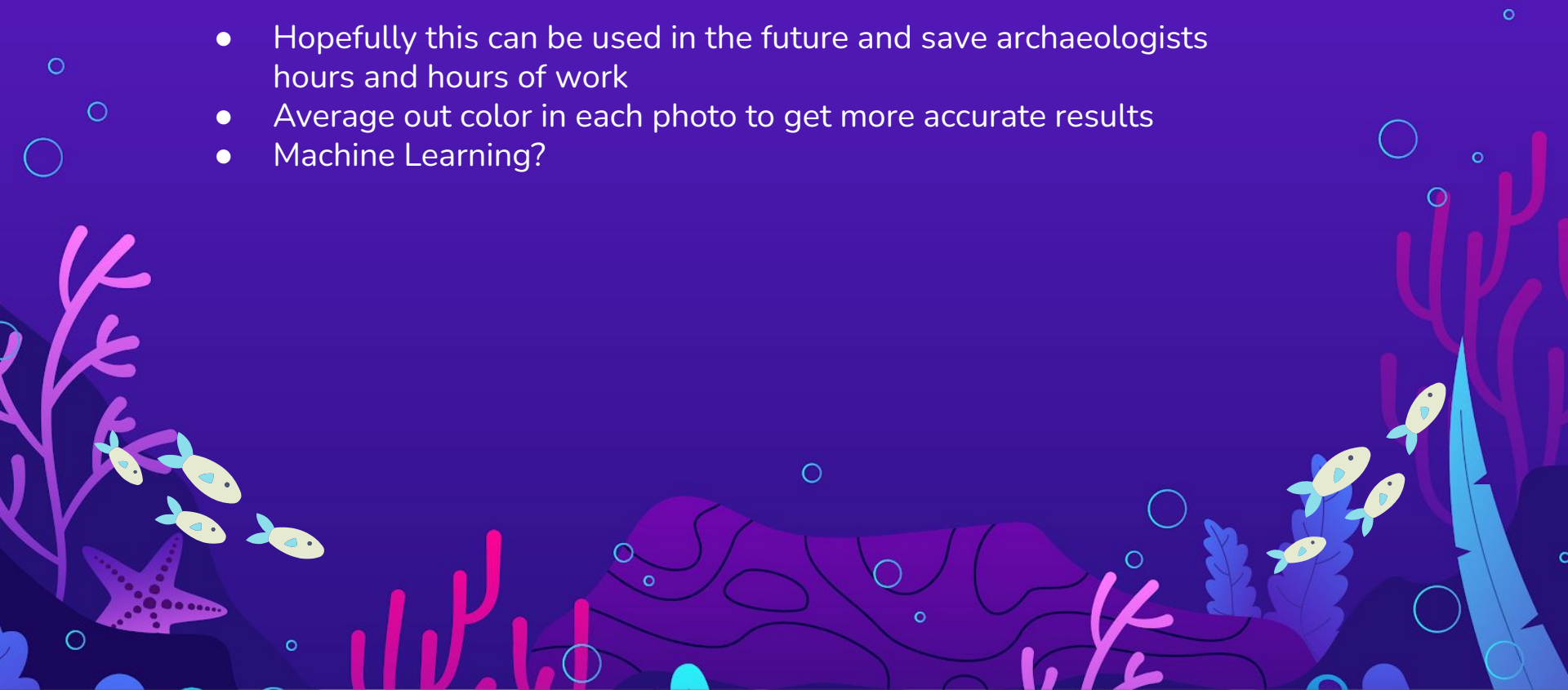
Results


Feature Extraction
0
200
400
600
800
0
200
400
600
800


HaBonim 2020
N
0 0.5 1 2 3 4 5 6 7 8 9

Future Work

- Hopefully this can be used in the future and save archaeologists hours and hours of work
- Average out color in each photo to get more accurate results
- Machine Learning?



Recap

- Learned how to use OpenCV and ImageJ
- Made many errors trying different methods like thresholding and edge detection
- Used a combination of gaussian blurs, changing the color spaces, thresholding, edge detection, and inversing to get my results
- Condensed this into a function that inputs an image and outputs a feature detected top plan