DETECTING GOLF COURSE BOUNDARIES

A COMPARISON OF COMPUTER VISION MODELS

Adam Cook, Brady Kane, Vivek Khanolkar, Kevin Mi, Ishan Patel, Vandana Prabhu, Rathziel Roncancio, Surya Salem, Deacon Shininger, Michael JY Son, Harim Song, Lauren Trinks

The Data Mine Corporate Partners Symposium 2021

ABSTRACT

Our research project involved isolating fairways of a golf course from the satellite image of the entire plot of land the course is on. We started by creating a general filter in Python that worked for the Purdue golf courses Kampen and Ackerman-Allen. After trying several edge detection methods, we found Canny edge detection to be the most effective algorithm. We then worked to generalize our method across all golf courses by tuning our parameters. To analyze the accuracy of our model, we manually annotated 700 courses and used intersection over union calculations to quantify the overlap. We believe our work on this project can help propel John Deere's understanding of automated boundary detection and enrich our knowledge as teammates and data scientists as well.

INTRODUCTION

John Deere is a manufacturing company that specializes in manufacturing agricultural, turf, construction, and forestry machinery. OnLink is John Deere's application to aid managers of golf courses with keeping track of tasks like mowing, watering, and fertilizing the course as well as aiding with equipment management and labor management. Our objective was to develop a model that will help improve current research around automated boundary detection for active golf course applications, specifically OnLink.

Golf courses occupy large parts of land in spots all over the world. They are composed of different components such as fairways, greens, sand bunkers, and water hazards. Thus, golf courses contain lots of grass and it is important to know where these different zones are on the course.

The satellite images of golf courses can help viewers determine where exactly the fairways and putting greens are, and the surrounding area of the golf course. We used the Google Earth Engine[1] alongside the National Agriculture Imagery Program (NAIP) dataset to find satellite images of both the Purdue courses as well as about 700 other satellite images of courses from all corners of the United States.

Figure 1: Raw satellite image of Ackerman-Allen golf course pulled

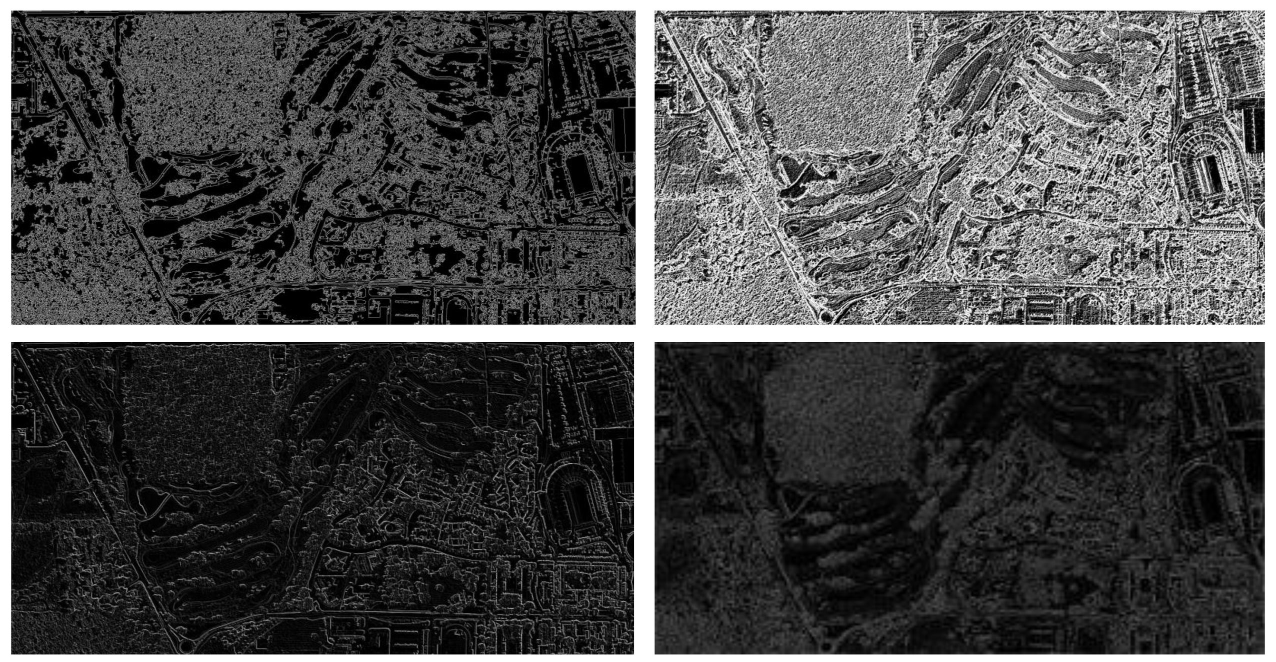

Figure 2: Comparison of Canny, Sobel, Prewitt, and Laplacian Kernel

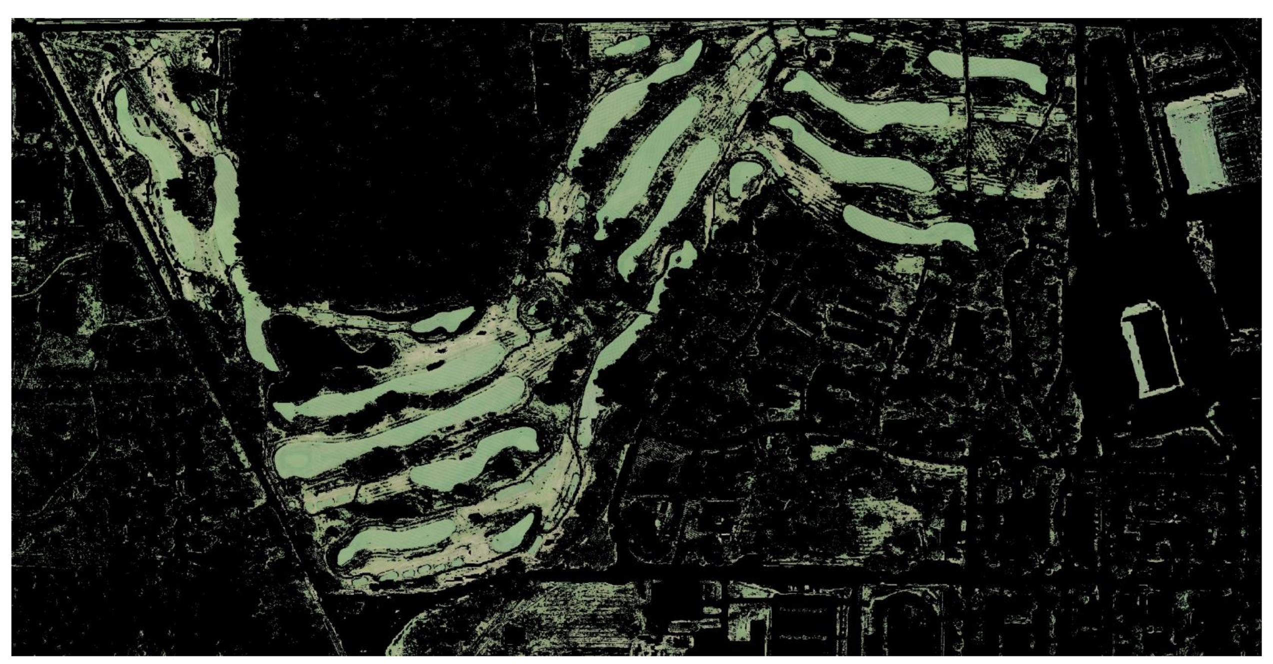

Figure 3: Image after HSV mask is applied

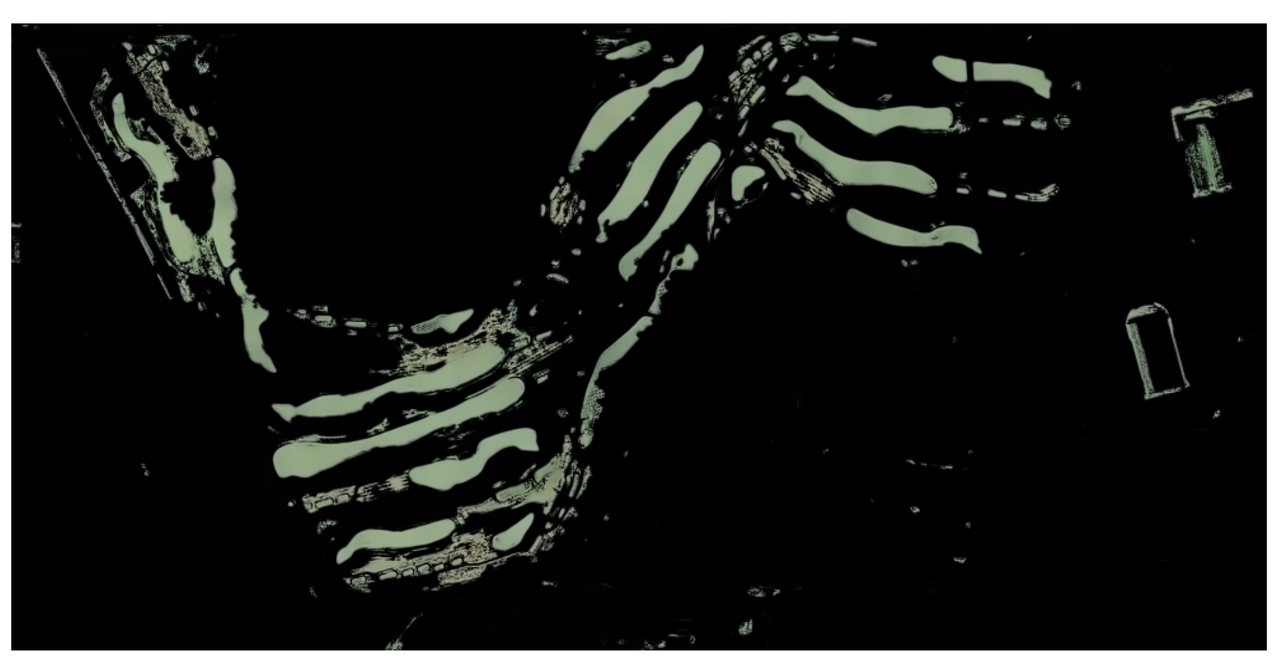

Figure 4: Image with no noise (non-local means denoising)

METHODOLOGY & RESULTS

STEP 1: GETTING THE IMAGES

- We found Google Earth Engine was the best way to obtain images. It had several options for satellite datasets with different resolutions and bands. The best dataset for our purpose was the NAIP dataset since it has the highest resolution at one meter per pixel.
- Once we had a way of obtaining the images, we had to transform those images to a more accessible format.
- The OpenCV package in Python was the best option. This package includes many edge detection algorithms and can take our image format as an input and output a standard format like PNG.

STEP 2: CANNY EDGE DETECTION

- We decided to use the method that worked best for the two courses at Purdue and then generalize it later. The most promising edge detection methods in the OpenCV package were Canny, Sobel, Prewitt, and Laplacian. Canny edge detection was the best.
- There was a lot of excess noise when using Canny, even when the parameters were tweaked, so we did additional processing.
- We used HSV masking to take an image and only keep parts of the image that fit a given color range.
- The color range was set to be the color of the fairways and the greens to isolate the golf course.
- To reduce the salt and pepper noise produced by the HSV mask, we applied a noise reducing filter, utilizing the technique 'non-local means denoising'.
- Then, we took the output of the HSV filter and ran it through Canny edge detection. The output is an image where only the edges of the course's fairway layout were left.

STEP 3: GENERALIZING OUR PROCESS

- The next step was to generalize the method to work with all golf courses.
  - We collected images of golf courses throughout the United States.
  - From there, we annotated the images using Makesense.ai[2] to get a 'ground truth'. Makesense.ai is a web tool that allows us to annotate our data with polygons, which will be compared to the Canny edge detection polygons.
- Next, we modified the process to be able to iterate through a folder of satellite images. We then used the contour method from OpenCV to correct a formatting issue we encountered with the annotated images.
- The next step was to write a Python program that would compare the results of this process to the ground truth. Our goal was to make a change to the process and see how much the change improved or worsened the results.

STEP 4: COMPARISON TO THE GROUND TRUTH

- We did multiple approaches for Canny, but we chose the intersection over union[3] (IOU) comparison for only one approach due to time constraints.
- IOU is a measuring tool to determine the accuracy of an object detector on a particular set of objects.
- We chose the approach for our comparison based on which method looked best visually.

Figure 5: Final annotated image via Canny edge detection

- We used IOU to measure the accuracy of our method (0-100%), which calculates the ratio of the area of intersection divided by the area of union.
- The average value was 0.22 (22%), which is primarily due to the high variance in colors for each satellite image.
- When tuned for a specific image, the value was much higher. For Ackerman-Allen, it was 0.44 (44%).

Figure 6: Intersection over Union results for Ackerman-Allen

CONCLUSION & FUTURE GOALS

Overall, we made great strides and learned so much in regards to golf course boundary detection. We learned how some Python packages will not work well together, and how to handle it using the Conda Python package handler. We discovered how much clearer the images we grabbed could be when we utilized the NAIP dataset. Through Canny we tested and tried numerous methods ultimately helping us reach new goals. Altogether, the final process we reached struggled to generalize to multiple courses, but when fine-tuned to one course, was more accurate. Throughout the course of the last two semesters, we pushed ourselves in a new problem space and believe we were successful in the progress we made towards the end goal.

If given more time, we would have liked to implement a data pipeline and a neural network that would have allowed us to make the processing of all our data, including golf course images, completed golf course image annotations, and image analyzation programs more adaptable to the user.

REFERENCES & ACKNOWLEDGEMENTS

[1]Google Earth Engine. (n.d.). Retrieved April 02, 2021, from https://earthengine.google.com/

[2]Make sense. (n.d.). Retrieved April 02, 2021, from https://www.makesense.ai/

[3]Rosebrock, A. (2020, April 18). Intersection over Union (IoU) for object detection. Retrieved April 02, 2021, from https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

Our acknowledgements go to Jarai Carter, David Glass, Ben Schwartz, Dr. Ward, and Maggie Betz.