Ivan Cano, Byungheon Jeong

Professor Schwartzman

DSC 180B A16: Satellite Imaging

March 9, 2021

From Rural to Urban: Tracking Urbanization

Introduction

Macro scale societal data, such as economic growth, urbanization rate, poverty rate, are by nature very difficult to gauge. As the sheer scale of the task makes it utterly impractical to actually count the elements of the population, we rely on surveys and sample based abstractions in order to generate reasonable approximations. This is not to suggest that these abstractions generate incorrect or wildly inaccurate results. In fact, many of these methods[1] are well-established and, given reasonable assumptions and careful execution, generate satisfactory metrics. However, there are a wide variety of ways for surveys to be designed and executed poorly, introducing confounding factors that muddle the results. Further, these surveys are often discriminatory towards those that are outside social-economic systems, as mail-based or phone-based surveys are usually only given to those who have the means to buy the infrastructure needed to receive the surveys. Lastly, corruption and incompetency has an especially outsized impact on this type of data as small issues easily cascade into massive errors.

These issues would not be much of a problem if the consequences of them were not so significant. Unfortunately, many of the most significant and consequential discussions are made with data based on these surveys. Economic development plans, infrastructure construction, political representation, governmental resource allocation, and such base their actions on the data in those surveys.

Thankfully, modern aerospace engineering, increases in computational power, and the open-sourcing satellite data allows for a population scale data collection and analysis. Such ambitions require ML in order to parse to the immense amount of data, models that must be trained on significant curated datasets in order to be useful.

Google Earth Engine (GEE) is one of these seminal technologies that democratizes this immense technical capability. GEE contains support and libraries to make complex analysis and visualization on geo-spatial data. Most importantly, GEE compiles 40 Petabytes of geo-spatial[2] data in an easy to utilize IDE. With these features, any researcher is able to access the results of images of multiple satellite missions over 40 years.

---

[1] Glasow, *Fundamentals of Survey Research Methodology,* 2005

[2] https://developers.google.com/earth-engine/datasets

However, we cannot simply use GEE data in making models or doing analysis without first labeling[3]. Unfortunately, there is no specialized labeling solution to Google Earth Engine that generates time-series data. That is to say that trying to use ui.label[4] to label many times in one geography over a period of time is a difficult endeavor. This capstone project directly seeks to improve this critical training pipeline, creating a service that will allow a quicker and less frustrating way to label large Google Earth Engine time-series datasets. We hope that this will encourage the proliferation of ML models in this space, so that decision makers and analysts can get better understandings of macro population data.

Data Loading & Modules

Napari

The most important part of our service is the visual labeling interface, or where the user actually runs does the labeling. It should not be too complex, but versatile enough to explore different parts of images and support multi-label labeling. To that end, we decided to use Napari, an open-source labeling interface that has extensive python API support.

Data Loading

It is good to have an excellent labeling service, but if there is no data to label, the service is rather useless. To this end, we have created scripts to help facilitate the download of geo-spatial data from GEE. It should be noted that this labeler is designed to specifically work with time-series, or data/images of the same geographic region over a period of time.

Data Structure

As the satellites collect data on a wide range of the electromagnetic spectrum, the data structure for a single picture (set continuous geographic area, set time value) will be a nxm (n = # of pixels in geographic area; m = number of bands used) matrix. However, as Napari (and any RGB based visual device) cannot display more than three dimensional matrix (nx3), we will also need a special “labeling” image that has identical metadata to the actual image. That is, the “labeling” image has to be of the exact same geographic region and the exact same time. Indeed, it would be helpful to view the “labeling” image as a reduced (three dimensions from m dimensions) version of the full image. Finally, the choice of these three bands depends on how well they serve to distinguish different components to the human (labeler’s) eye.

Program Process

In a high level view, the program is a pixel-wise labeler. It works by displaying the dimension-reduced “labeling” picture on the visual interface (Napari) so that the user can draw polygons that circumscribes an area of interest, then uses those polygons to choose the pixels in the full image and puts those pixels in a numpy array. Alongside that array, an “label” array that corresponds to each pixel value holds the label. These two arrays can be used directly by pytorch.

---

[3] Admittedly, there are many valid ways to go about unsupervised learning and such exploration may certainly yield interesting and useful results. However, labels are still needed to make sense of things found by these algorithms and generally, supervised learning is more useful when dealing with geo-spatial analysis tasks

[4] https://code.earthengine.google.com/08146a0183f51ff94a34bb90ae987e0d

Import modules
Here is the general flow of the function in geel.py:

For each image in dataset ->
Label the "labeling" image
Extract Polygons
Get pixels of the full image that are in the polygons
Input label into label dataset alongside the input of pixel in the data dataset

Additional Features

While running through the dataset, there may be times where Napari crashes or there are reasons to terminate the labeling session. To avoid a memory wipe of previous work, there is a checkpoint system that saves the pixel information and the connected label information in its numpy arrays. Further, there is a file checkpoint system that stores images that were already labeled or skipped.

Pre-Labeling, Napari Pre-Labeling, Napari

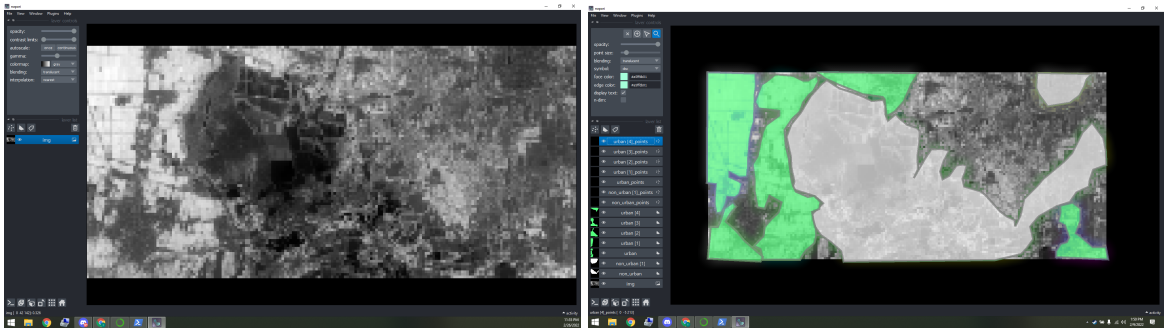

Figure 1. Xiong An New District, taken September 28, 2021. The above images represent the visual representation of our Xiong An New District in the Napari labeling service, before labeling and after labeling.

Data

The data we collected was from the landsat 8, a satellite image repository created through the joint collaboration of NASA[5] and USGS[6], through their Landsat 8 satellite program. Within the Landsat 8 image repository we used all available dates, which included images spanning from March 18, 2013 to January 13, 2022. The specific region that my team settled on using for our modeling and processing was that of the Xiong An New District, and the Xushui District in Hebei China. We chose these regions because of the expansive research that has been done in China using landsat satellite images. As this project took considerable research to implement many of the techniques we utilized in this research paper, the region in Hebei seemed liked the perfect place to collect data from, as we were able to continuously compare our processes and outputs to check if they aligned with those of other contemporary research papers working satellite image processing. Moreover, as Xiong An New District is a rapidly developing region in China, being one of the chief administering districts in its area of Hebei, my team suspected we could find some promising results that have not yet been found in this specific region. As for Xushui District, even though it does not have any political or economic importance in the region

[5] National Aeronautics and Space Administration
[6] United States Geological Survey

of Hebei, it was chosen because of its proximity to the Xiong An New District. For the sake of this project we incorporated data from Xushui District to be able to function as a test set to test the effectiveness of the models we developed from the Xiong An New District data.

Utilizing methods from google earth engine, we were able to create a bounding box for our desired region to then pull our images in the form of .tif files from both the Xiong An New District, and the Xushui District each of which consisted of 89 by 189 pixels in length and height. Although the data was originally pulled in as a two dimensional object, for the majority of the project we worked with this dataset as a three dimensional object as they later on gained a new dimension in the form of bands.

Data Cleaning

As helpful as the landsat 8 repository was in providing our team with valuable satellite images for our projects, it also brought with it many of the complications inherent to the data itself when working with the images. Landsat 8, like many of its satellite antecedents suffers from scan line correctors that serve to compensate for the forward movement of the satellite as it orbits the earth. Unfortunately, the instruments that Landsat 8 uses to correct for the discrepancies in the satellite's movements move in an oscillating vertical pattern, many of the images we use are highly prone to long strips of black lines throughout many of the images we collected. Unfortunately there are no great inherent solutions to filter out the images that were highly affected by the scan line correction so to filter out the most affected images we simply manually picked the worst images that we had downloaded. This process was as simple as collecting the dates of the images selected and filtering out the images by those dates of the ones that were most affected by line correction, so as to be excluded in the final downloads of our .tif files.

Some other quality issues we faced with the data were those related to meteorological disturbances in the images. Many of our images were prone to being badly captured because of clouds and other weather related obstructions. To solve this issue my team implemented a cloud filter in our data collection process. As the images we worked with had cloud values associated with how much meteorological interferences were present at the moment of their capture, we used this cloud score present within them to be able to filter out the worst scoring images (those images that had a significant presence of clouds within them).

Data Processing

After completing the data cleaning portion of our project, we then moved on to data processing. In order to extract valuable data from our images, we had to make use of some of the features inherent to Landsat 8. Landsat 8 provides a variety of bands that come included in all of its images. Some of the different techniques that Landsat 8 utilizes in extracting values from its images is through the use of electromagnetic bands. Bands B1 through B11 all relate to images under electromagnetic wavelengths, ranging from coastal aerosol detection bands to high-gain thermal Infrared wavelengths. Although what is most important to note here is that they all gather data on specific regions of the electromagnetic wavelengths. Using the aforementioned B1 band as an example, its specific range within the electromagnetic wavelengths captures scores of data between 0.43 - 0.45 μm, and as you go up in the bands that detect wavelengths, their wavelength detection range also goes up. For the purposes of our project we decided on using all

of the bands related to the electromagnetic spectrum so as to input a wide range of data into our model so that it could evaluate the weight of each of the data points related to the images.

Some additional bands found within the Landsat 8 dataset that we decided to incorporate into our test images were those of the solar projection bands. The solar projects bands which were the bands that included SAA, SZA, VAA, VZA, are all bands related to the shadows projected by objects, each of which varies depending on the angle and time of the sun's projection of the sun's center onto the horizontal planes. As our projects focuses on Urban expansion, and human made structures like building and city infrastructure tend to stick out of the ground, the majority of our regions of interest could be easily captured by the shadows they project. As such, the values outputted by these projections were predicted to provide us with highly valuable information to track the growth of urbanization. Furthermore as these are projection of the sun onto the ground it add some qualities of a 3-dimensional plane to our dataset as these values capture that is projected by the height of objects at a certain point in time, and could thus provide our models with a richer depth of information by which to make more accurate labeling predictions.

Lastly, the final band that we incorporated into the test set was that of band QA_RADSAT. Band QA_RADSAT produces values based on a monotonically increasing function whose range is restricted by the maximum irradiance, where pixels whose intensity corresponds to this maximum are known as saturated. As this band captures the reflected energy per region, our team believes that it could capture different levels of energy between the urban and non_urban regions as both areas of interest would be made up of vastly different materials; one made up of mainly concrete and the other of mainly earth and vegetation. Aside from the inclusion of the previous reasons stated it could also provide further diversity for our model to train on.

$$NDVI = \frac{SI_{Band\,5} - SI_{Band\,4}}{SI_{Band\,5} + SI_{Band\,4}}$$

Formala. 1. NDVI bands, where SI represents the satellite image processed.

In conjunction with the images we processed using the 13 bands of electromagnetic, solar projections, and energy reflection, we also processed a second set of images under a NDVI band to be able to produce an image that we used for the labeling portion of our project. Our team decidedly used NDVI for the excellent differentiation that it provides between urban and non urban regions of an area. When comparing it side by side to other native bands in landsat 8, it stands out for having clear lines distinguishing its different geographic regions in intense contrasting colors. In its visual representation, its urban areas become highlighted in neon green while its non urban regions become a dark shade of blue, providing an excellent color and shading scheme to distinguish the two regions by. Unfortunately, because our Napari service only displays .tif images through monochromatic visual representations, the final version of the images we labeled are represented in black and white but, because NDVI records its visual representation of its .tif files in contrasting light green and dark blue, it has the distinct advantage over other bands that are shown in 3 colors, perfect for the visual representation that Napari provides us. When interpreted by the Napari the light green that represents urban settlements,

turns to white and any vegetation which is dark blue is represented in black, and shades of grays can be interpreted as having different levels of vegetation and/or urban settlements within it.

Band Name	Pixel Size	Wavelength	Description
B1	30 meters	0.43 - 0.45 µm	Reads in Coastal aerosol
B2	30 meters	0.45 - 0.51 µm	Reads in Blue light.
B3	30 meters	0.53 - 0.59 µm	Reads in Green light.
B4	30 meters	0.64 - 0.67 µm	Reads in Red light.
B5	30 meters	0.85 - 0.88 µm	Reads in Near infrared wavelengths.
B6	30 meters	1.57 - 1.65 µm	Reads in lower µm shortwave infrared wavelengths
B7	30 meters	2.11 - 2.29 µm	Reads in higher µm shortwave infrared wavelengths
B8	15 meters	0.52 - 0.90 µm	Reads in Panchromatic wavelengths
B9	15 meters	1.36 - 1.38 µm	Reads in Cirrus wavelengths
B10	30 meters	10.60 - 11.19 µm	Reads in low-gain Thermal Infrared wavelengths. This band has expanded dynamic range and lower radiometric resolution (sensitivity), with less saturation at high Digital Number (DN) values. Resampled from 30m to 100m
B11	30 meters	11.50 - 12.51 µm	Reds in high-gain Thermal Infrared wavelengths. This band has higher radiometric resolution (sensitivity), although it has a more restricted dynamic range. Resampled from 30m to 100m.
QA_RADSAT	30 meters	NA	A monotonically increasing function whose range is restricted by the maximum irradiance.Where pixels whose intensity corresponds to this maximum are known as saturated.
SAA	30 meters	NA	Pixel values generated by shadows casted by the Solar Azimuth Angle of the day.
SZA	30 meters	NA	Pixel values generated by shadows casted by the Solar Solar Zenith Angle of the day.
VAA	30 meters	NA	Pixel values generated by shadows casted by the View Azimuth Angle of the day.
VZA	30 meters	NA	Pixel values generated by shadows casted by the View Zenith Angle of the day.
NDSI	30 meters	0.60 - 0.90 µm	A infared like band that generates values based on the formula:

Figure 2. A collection of all the bands and their associated pixel size readings, electromagnetic wavelength, and description of their values.

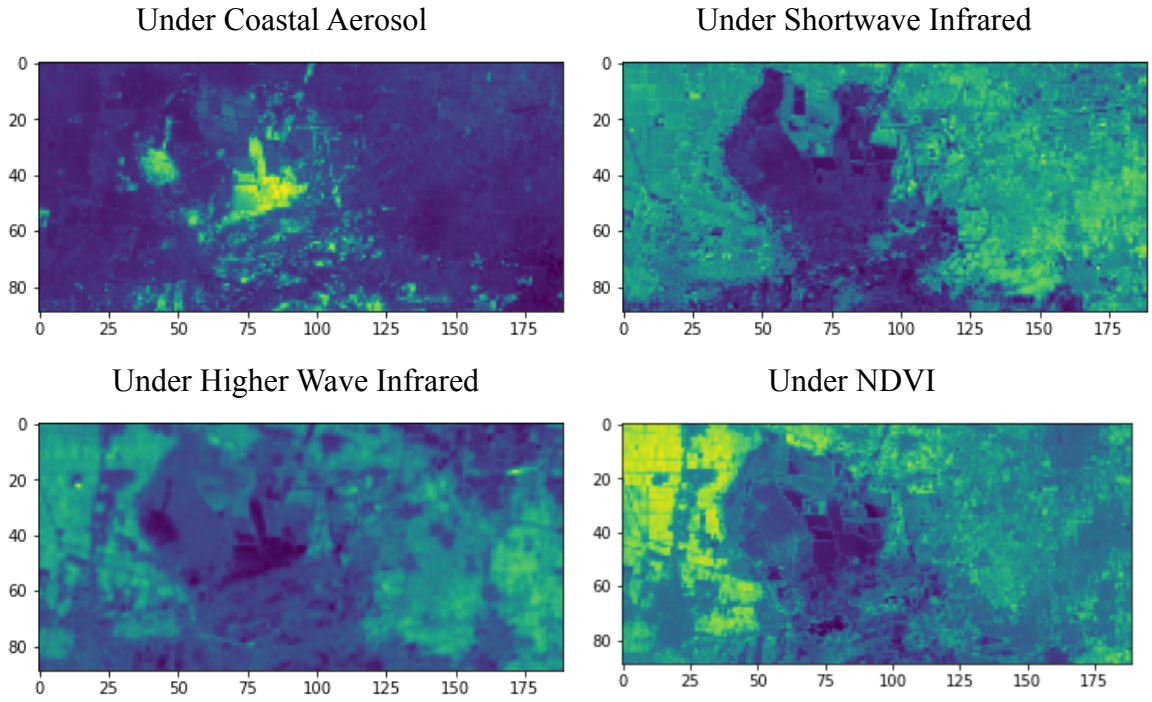

Figure 3. The above images represent an image from the Xiong An New District, Gissar Range, takenSeptember 28, 2021, under different bands that were tested. Top left sector contains the test image under a coastal Aerosol, (B1). The top right image contains the test image under a shortwave infrared band, (B6). The bottom left sector contains the test image under a higher wave Infrared band. The bottom right sector contains the test image under a NDVI band.

Modeling

After performing all the procedures related to the data cleaning, processing, and labeling of the data, we had at last obtained the data by which we could train our model on. This step of our project was performed intuitively and straightforwardly. We tested a variety of different classification ml models including logistic regression, and a variety of decision trees, and after testing their performances of the different ml models we had decidedly chosen to use a random forest classifier based on its performance based results. Once we had selected our ml model we began hyperparameter tuning, testing the most relevant parameters in our ml model including max_depth, n_estimator, max_features, and min_sample_leaf. For each of these parameters, we opted to individually test them to test their performance based on a wide range of values. Once we had models for each of the parameters we imputed we subsequently tested for their accuracy and recorded their highest performance input. When all parameters were tested, all the recorded parameters were then inputted into our last model, to be the basis for our label predictions.

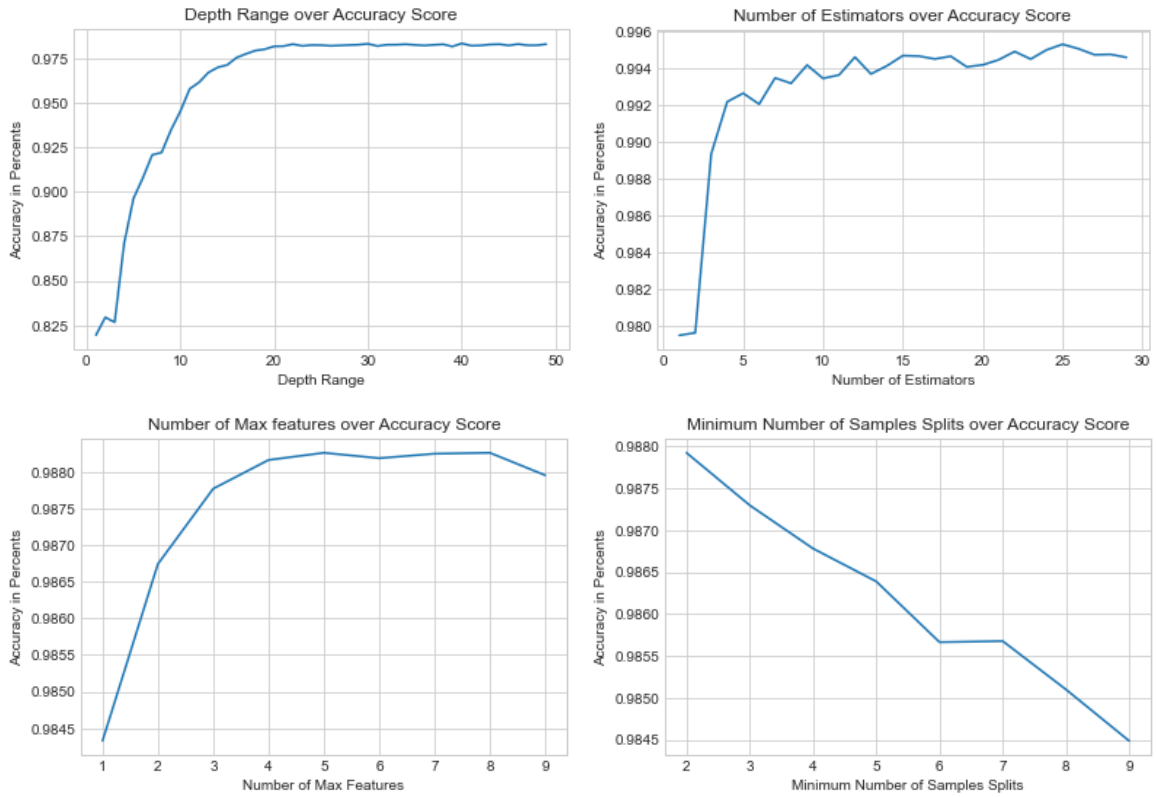

Figure 4. The above graphs represent our ml model under different parameter inputs, each respective of their own tested x value. The top left graph represents accuracy over depth range. The top right graph represents accuracy over estimators. The bottom right graph represents accuracy over the number of max features. The top left graph represents accuracy over the minimum number of sample splits.

After testing our model with our test set, we recorded an impressive 98.7% accuracy in being able to correctly label our images as either urban or non urban.

Analyses:

$$Urban\ Score\ = \frac{count(urban\ labels)}{count(non\_urban\ labels)}$$

Formala. 2. Urban score, where count is a function x equal the occurrence of the input

With our model built, we obtained the means by which to analyze the overall trends of our regions of interest, Xiong An New District, and the Xushui District in Hebei China. With the model built we were able to input the entirety of labeled data of the Xiong An New District and track its trends over time. To accomplish these means we created a simple urban expansion scoring unit, found in formula 2. By calculating and tracking the urban score over time, we were able to quantifiably observe the relation that urban expansion would have over time. When graphed we observed a gradual increase in urban score over time. Quantifiably this observation translated to have a positive correlation coefficient of 0.0554, and having standard deviation between the urban scores of .921. Although this might seem like a weak positive trend between

urban expansion over time, one must consider that urban expansion overall is a slow process that takes time to grow. All things considered this positive trend between the two variables seems aligned with the expectations of our group. When the data is observed over time in days we can note that our data translate to an increase of 0.000236 urban score, which translates to a .086 increase of urban score every year. In the grand scheme of our data our findings translate to finding an increase of 0.755 urban score between April 10, 2013 to January 1, 2022. Overall, these substantial results were found to indicate that there was an overall increase to urban score and therefore to urban expansion in the region of the Xiong An New District.

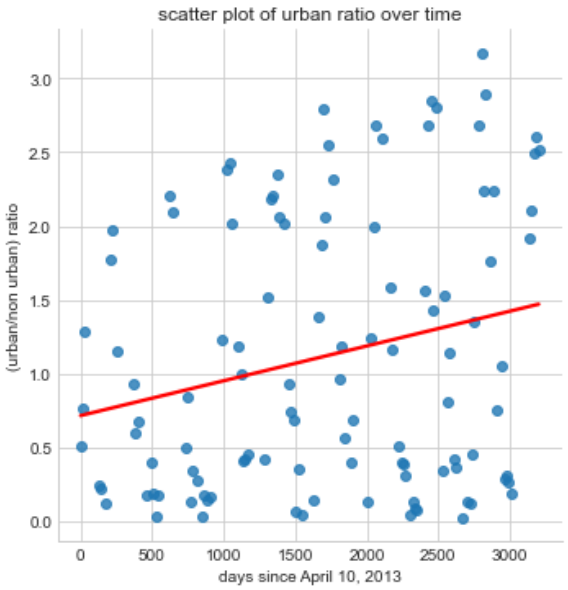

Figure 5. A scatter plot of the urban ratio over time, in terms of days past since April 10, 2013, with accompanying linear regression line.

Delving a little deeper into the data my team wanted to explain why the standard deviation between the data points was so high, after all an std score of .921 seemed abnormally high, and we wanted to explain why that was. After plotting the urban score over time again, it was found that under a line graph, the data points seemed to have very distinct oscillation patterns within them. Even when accounting for the correlation coefficient of the data, and detrending the relationship of time over urban score, the oscillation patterns stayed consistent. Moreover these oscillation patterns seemed to follow a consistent pattern that reached its lowest point, in terms of urban score, in the Hebei's warm weather season between May 10 to September 19, and its highest point in the Hebei's cold weather seasons between November 25 to February 24, 2022 (weatherspark)[7]. When averaging out the data, by months the graphical distribution of the data proved to be consistent with the seasonal oscillation weather patterns of temperature recorded over months in the Hebei province in China.

[7] https://weatherspark.com/y/129994/Average-Weather-in-Hebei-China-Year-Round#Figures-Temperature

Our team suspects that this oscillation could be linked to these changes in temperature and weather that change depending on the seasons of the year. This would make sense as our use in the NDSI band to label our image was intended to differentiate between vegetation and urban areas in our data. Therefore it would also be logical to conclude that much of the labels that were classified non_urban areas would also be highly dense in vegetation as these would be the regions that would be most visually differentiable and would thus have been classified as non_urban. As such, our non_urban label variable would be highly susceptible to seasonal changes such as in weather, especially temperature. And although we do not have quantifiable evidence to support our hypothesis on the seasonal shifts in our data, as we are limited in time for this section of our research, this theory does pose interesting questions to be solved if we were to ever continue expanding upon this project.

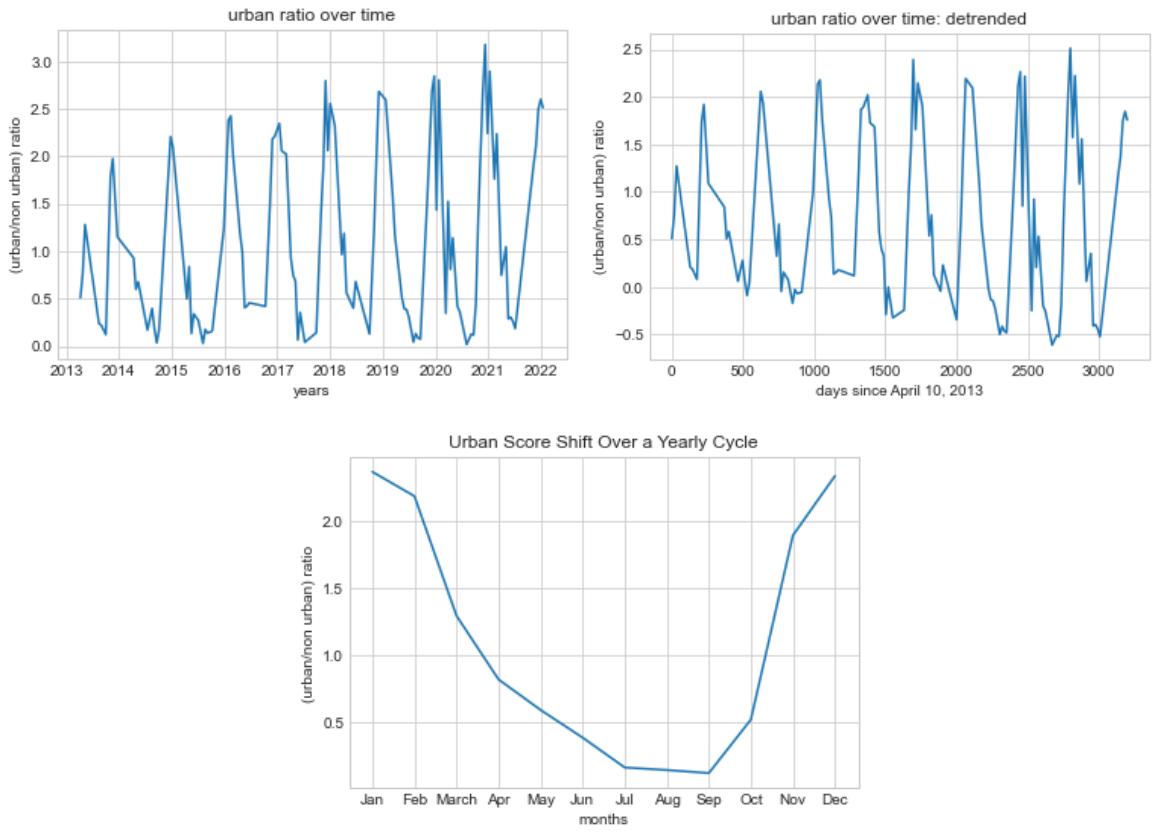

Figure 6. Top left is a line graph of urban sore over time in year. Top right is a line graph of urban sore over time, detrended, in days since April 10, 2013. Bottom is a line graph of the average urban score over months of the year.

Limitations

Even though our model proved to return a relatively high degree of accuracy for one of our regions of interest, Xiong An New District, it should be noted that our models and our predictions do not reflect the application of our labeling service in all geographic regions. Even within China, there may be cases where our model's performance may result to be considerably worse, and in cases like these it becomes difficult to even conduct any analytical work to be able

to quantify any trends if present. This same limitation occurred several times throughout our project, such as Uyo, Nigeria where because of the different building material, and the scarcity of cement in developing Nigerian areas, the classification of urban settlements became extremely difficult to track using the techniques that we implemented in this project. Even within regions of China, accuracy for our classification model was lower than our team would have liked in certain regions. Most notably, when attempting to apply our model to the Xushui District, in China, it was found that the accuracy was too low to be able to conduct any type of analyses, as the data gathered might have been too inaccurate to gather reliable information from this district.

Conclusion

In our exploratory analyses our team was able to discover positive correlation trends through our implementation of our ml models to predict regions classified to be either urban or non_urban. By observing the change in ratio of urban regions to non urban regions over time our team was able to observe the relations of urban growth over time. Although the correlation coefficients did not seem to provide much of a positive trend based on the raw statistical values provided, in the broader context of urban expansion over time, the data seemed to predict urban growth proportional to what could possibly be expected from a developing urbanizing region. Furthermore, when analyzed under the context of fluctuating data under seasonal trends the variance found in our data was to be expected using the methods that we implemented to label urban and non_urban areas, in regions that would contain high densities of vegetation. All things considered, the results we found did seem to align with our expectations. If given the opportunity though, we would definitely explore the trends we found more indepthly and apply what we have discovered to change the way we approach our labeling system to improve upon our results.

Conclusion on Code

Our infrastructure presents a substantial improvement in labeling google earth engine images that are in a time-series format, or images of the same geographic area over a period of time. With regards to our development plans, we have met all of our initial goals. Our goals were the following:

1. Integrated Napari to enable pixel-wise labeling and to output numpy arrays
2. Iterate through a dataset that consist of reduced “label” images and the full images
3. Support arbitrarily large numbers of label types
4. Develop download scripts that are easy to use
5. Deploy script on cloud instance so service can be embedded in website

We have fully succeeded in the first three goals, partially succeeded in the fourth, and were not able to accomplish the fifth.

Nevertheless, the program is “complete” and the fourth and fifth features are to maximally reduce friction and set-up complications for users. We aim to add features when we have additional time. Further, Napari has some quite advanced features that we were not able to fully explore in this project. As such, we hope to explore the usage of these features to add additional functionality to our service.

Lastly, the true "golden fleece" of this project would be deployment onto a public cloud in order to integrate the data exploration, download, and labeling into one platform/website. Due to the fact that Napari uses the Qt5 OpenSL package, the package was too large for a browser OS, we were not able to run it on web-based applications. There may be other visual interface solutions possible, which may better facilitate cloud integration.

Citations

"Weatherspark.com." *Hebei Climate, Weather By Month, Average Temperature (China) - Weather Spark*, https://weatherspark.com/y/129994/Average-Weather-in-Hebei-China-Year-Round#Figures-Temperature.

Bühler, Fabian, et al. "Process and Economic Optimisation of a Milk Processing Plant with Solar Thermal Energy." *Computer Aided Chemical Engineering*, 2016, pp. 1347–1352., https://doi.org/10.1016/b978-0-444-63428-3.50229-0.