Final Project

Title: City Prediction Model
Notice: Dr. Bryan Runck
Author: Alexander Edstrom
Date: 9/19/21

Project Repository: *GIS5571/Final at main · AlexanderEdstrom/GIS5571 (github.com)*

Abstract

The focus of this project is on creating a model that tracks the expansion of a city over a given time interval and then utilizes Machine Learning to create a prediction of future expansion. The overall goal of this model is to be implemented within an application that fully automates the process by doing all of the intermediate steps within the scope of the application, rather than having to do the steps manually. Ultimately I want this application to only take RGB images as the input and be able to output the prediction. With that in mind, this project has been developed with that “RGB image only” data limitation to allow it to be implemented into the application in the future. For example, in this paper I only use Sentinel 2 imagery to create the classification models and to train the CNN model. The classification method used to create the hotspot result verification found within the “Results Verification” section uses all bands of the Sentinel 2 images to reflect the most realistic results someone would reasonably come to if doing hotspot analysis alone. The classification of arid areas often involves extra layers of data that are not necessary to include in more heavily vegetated areas (Gaur, et al., 2017). Helpful statistics to include within arid classification are things like surface temperature, air humidity, soil humidity, and precipitation (Perez-Aguilar et al., 2021). With that in mind I wanted to center this project around data from an arid city, leading to the selection of Phoenix, Arizona.

Introduction & Problem Statement

Within the realm of intelligence analysis one of the most common pieces of data to have access to is satellite imagery. Being able to extract as much information as possible out of images alone is the basis of multiple GIS analysis methods, classification alone is used in a large swath of common analyses. Currently, prediction of city growth is often based on a wide variety of data and statistics. Data such as historic growth rate, population growth, land cover change, and economic growth are often used to predict future growth of a city (Hussein, et al. 2014). In the case of foreign intelligence analysis, many of these common data points could be unknown. To combat this data limitation, machine learning can be leveraged to derive as much information out of an image as possible by training a model to recognize what parts of a city are likely to go through change in the near future based on images alone. The benefit of using machine learning is that it will pick up on relationships and patterns that humans would never be able to recognize with traditional analysis methods. That aspect is both a pro and a con, as it is impossible to see exactly what patterns the model is basing its predictions off of, it is a black box. These general project requirements are outlined within Table 1 below.

Outside of intelligence analysis, this model could easily be used in tandem with other methods of growth detection. While it's unlikely that this model will be as accurate as predictions made based on more abundant data as stated above, this can always be added to other analyses. Understanding how a city grows and changes provides the important context needed to guide many decisions within government and industry. Sectors like Public Works and Commercial Development could use growth information to decide how and where to construct roads, buildings, and utility hubs.

Table 1. Requirements

#	Requirement	Defined As	(Spatial) Data	Attribute Data	Preparation
1	Satellite Imagery	Raw imagery from Sentinel 2 of Phoenix, Arizona	Raster Image		Divided into cells for CNN Model
2	Image Classification	Classified version of image (land cover)	Raster Data	Landcover	Classified using unsupervised pixel based iso cluster
3	CNN (Machine Learning Model)	Pre-existing Convolutional Neural Network model			Model is retrained with Sentinel 2 Phoenix images

Input Data & Study Area

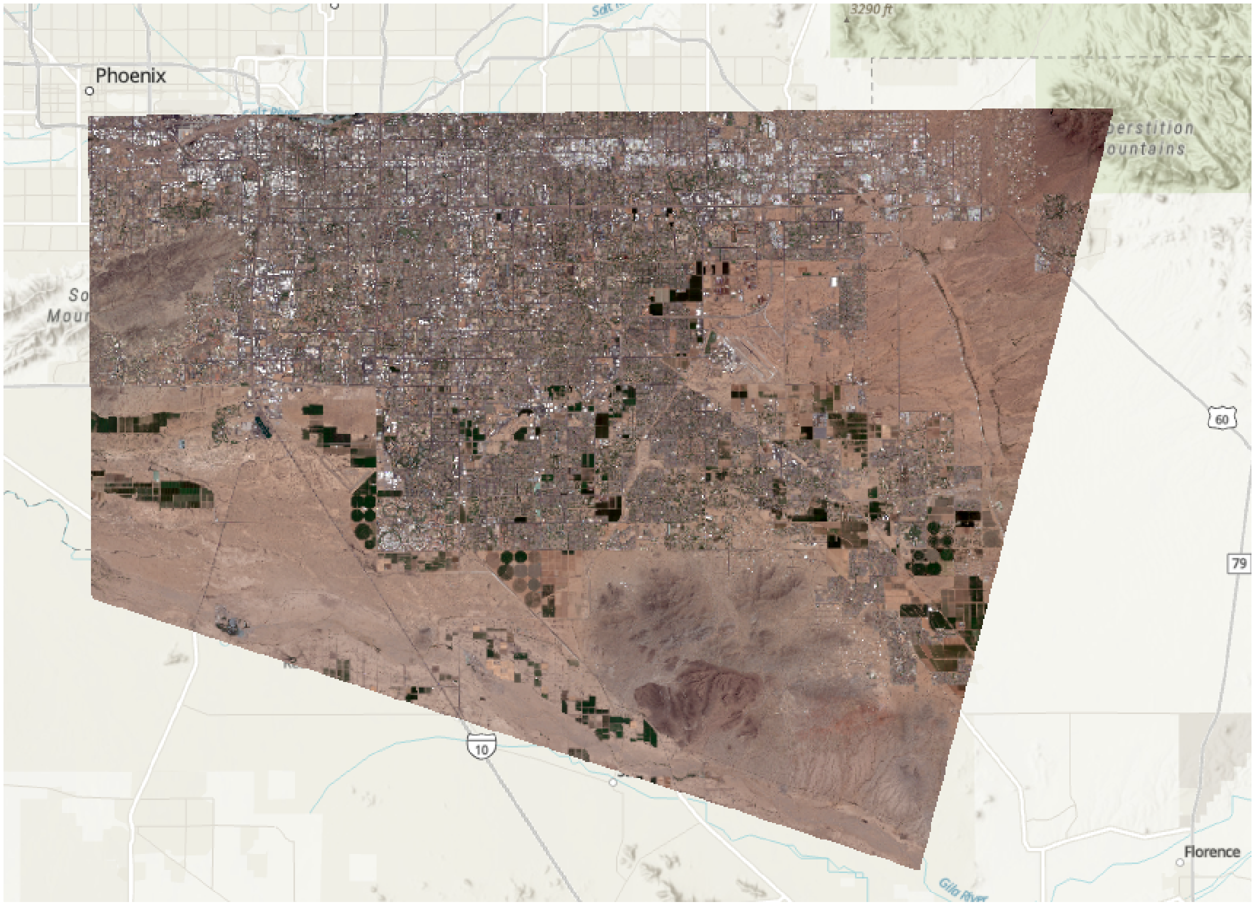
The study area consists of the south eastern edge of Phoenix, Arizona. The full extent of the input data is shown in figure 2 below. The polygon used as a mask to clip the study area is a portion of a Sentinel 2 image covering the entire city of Phoenix. The image is clipped down as to only contain the part of the already developed city, the area right on the fringe of the development, and natural area that has not yet been developed. This is done to ensure that the input data represents each of the developed, developing, and not developed categories. The representation of these categories is important for the training portion of the CNN model. The Sentinel 2 imagery is acquired from USGS Earth Explorer via the GUI, however a data ETL pipeline is included within the code. The ETL is excluded from being used in the actual analysis done within the scope of this project as USGS restricts its API access and the request submitted for this project was denied. The structure of making requests is public however, so an API pull request can be written, just not executed. In total. 2 images are used in this project, both Sentinel 2 images, both covering the same extent, both clipped to the same mask, both with less than 5% cloud cover, one of the images from early July 2017 and the other from early July 2018. The Sentinel 2 images come in .tif form with 13 bands.

The only other piece of data used in this project is the pre-existing CNN model that is retrained with images mentioned above. The model chosen is a multipurpose image classification convolutional neural network. It is designed to be retrained on relevant images. This model is found within the TensorFlow library and runs based on the TensorFlow python package. More information about the data is found in table 2 below.

Table 2. Input data

#	Title	Purpose in Analysis	Link to Source
1	Sentinel 2 Images of Phoenix, Arizona	Raw input dataset for classification, change detection, and to use as training data for the CNN..	USGS EarthExplorer
2	Efficient V2 XL 21k	CNN model from TensorFlow used for growth prediction	Efficient V2 XL 21k

Figure 1. Study Area (2017)



Methods

The general workflow is outlined in figure 2 below. First, the 2017 and 2018 imagery is retrieved from EarthExplorer. As stated above, the API is restricted so the GUI has to be used. The .tif images are then sent through an unsupervised pixel based iso cluster classification to classify them into pervious and impervious land cover. All 13 bands of the Sentinel 2 data are included to perform this classification. Permeability is the sole metric used to define whether or not the given land cover is developed. Impervious surfaces are classified as developed. Then change detection is performed on the two classifications. The change detection raster only contains the pixels where change occurs and is classified based on what kind of change happened (Undeveloped to Developed OR Developed to Undeveloped). After classification, the two images are cut into cells of size 512x512 pixels. This is the resolution that the CNN is set to train and predict based on. 512x512 equates to about 25 km^2 of area. This size was chosen to make each cell both large enough to keep computing requirements low and small enough to provide a useful prediction size. Cells too large would reduce the usefulness of the prediction as they would highlight too large of an area as change or no change. After the images are divided they are converted into .jpg images containing only the RGB bands, this is an intermediate step that needs to occur before the images get fed into the CNN for training. The .jpg cells of image 1 (2017) are then compared against the change detection raster to be separated into one of two buckets, “Change” and “No Change”. This is done quantitatively by looking at the change detection raster statistics of each cell and calculating a percent of the Undeveloped to Developed class. Cells that contain 12% or greater Undeveloped to Developed pixels are placed in the “Change” bucket. 12% was chosen as the cut off as it represents three times the average concentration of Undeveloped to Developed pixels. This will highlight the areas where growth/development is happening at a greater rate. A visualization of the cell divisions overlaid on the change detection is shown in figure 3. Once the cells are classified they will be fed into the CNN to act as training and validation data. A new model is not being created specifically for this project, the concept of transfer learning is leveraged to simply retain an existing image classification model with data relevant to this project. The model separates the classified input data into both training data to learn from, and validation data to check itself. As the model trains it records the progress and accuracy to later be graphed. Once the training and validation is complete the model can be applied to the 2018 image that has not yet been seen by the model for training to give a prediction for each cell. The model will predict if change will occur in the cell by the next time step of 1 year due to the fact that it was trained on data based on the change detection from 1 year in the future.

Figure 2. Dataflow Diagram

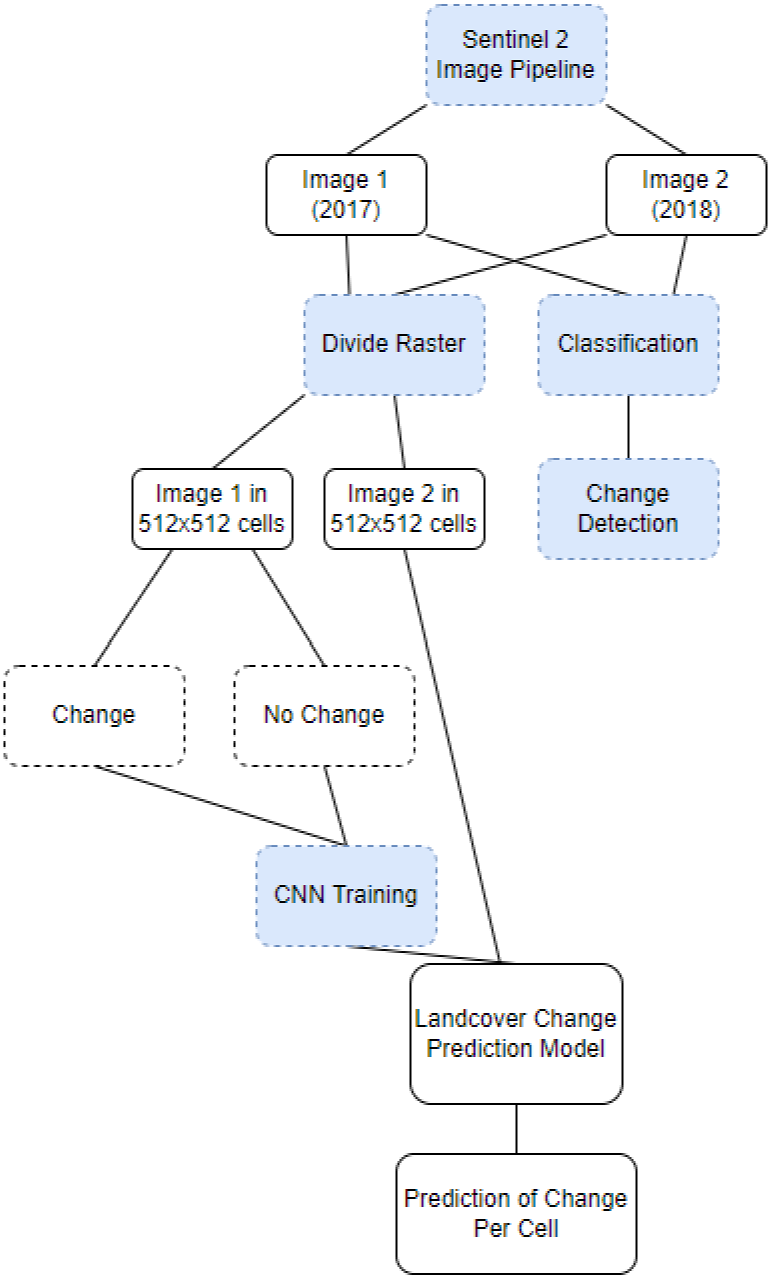
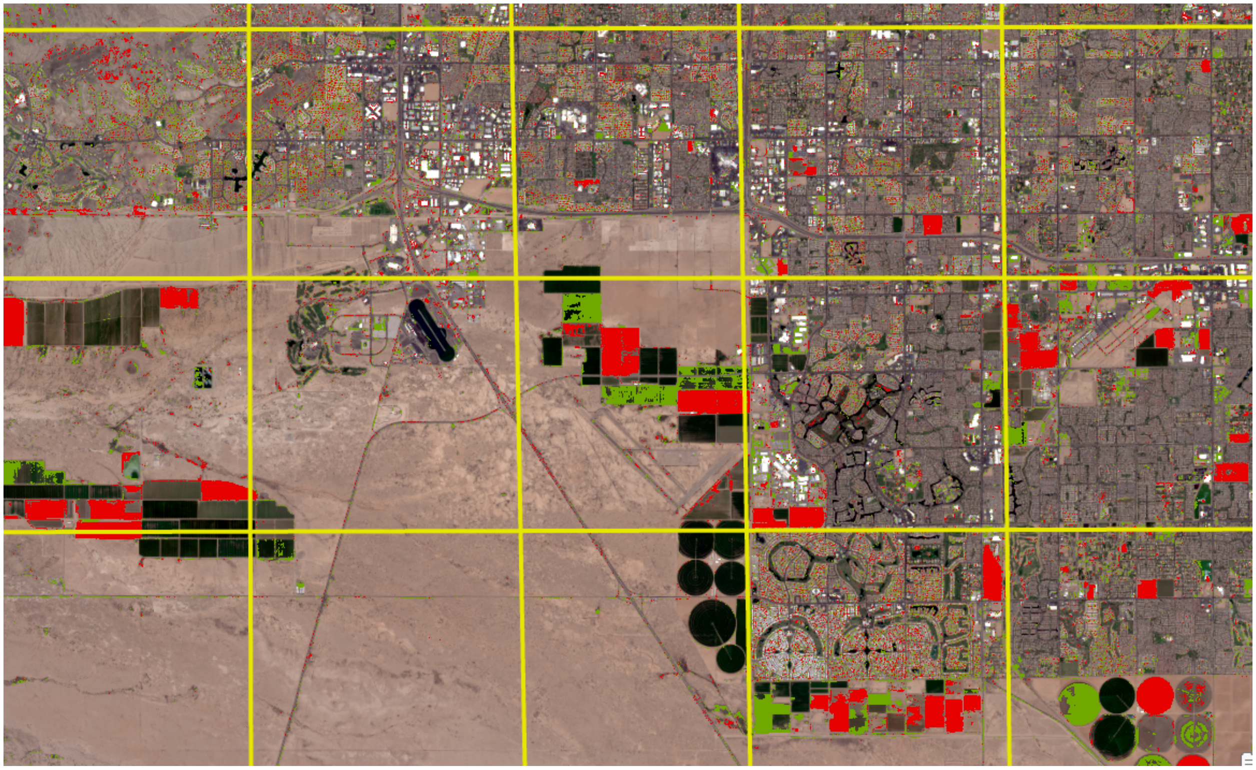


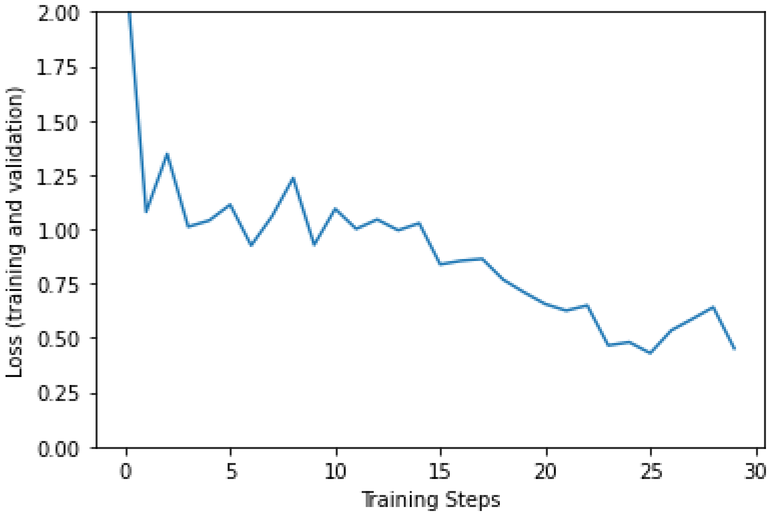
Figure 3. Visualization of 512x512 Cells and Change Detection (Undeveloped to Developed shown in green)

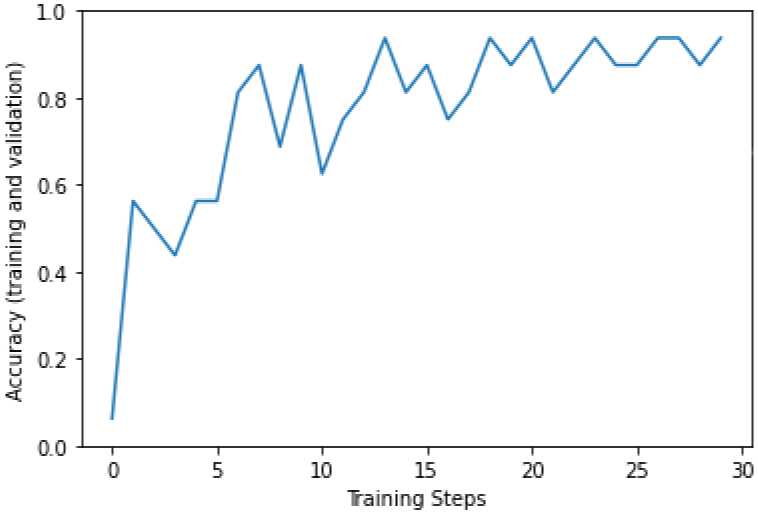


Results

Figure 4 shows the loss and accuracy of the CNN's training over time. The training session shown in these graphs used only a portion of the 2017 image (about half) as the training. Another quarter of the data was set aside for validation. This was done to address a memory issue, this is further discussed in the discussion section. The training graph shows that the model was able to reach around high 80% to low 90% mark. An average of 5 runs showed an accuracy rate of 88% and a loss of .58. Each run used 30 epochs and covered a total of $750km^3$ of data. Alternative result considerations and comparisons are explored within the validation section.

Figure 4. CNN Training





Results Verification

The accuracy of the CNN is tracked and reported by the model itself shown in figure 4.. Both the loss and accuracy are recorded at each epoch of training and graphed after the training is complete. The basis of this accuracy calculation is comparing the predicted class to the know class of each cell. Once the training data is run through and has been used to train the CNN, the validation data can be used as testing data, as the model has not yet seen that portion of the data. With an accuracy rate of 88% these results are certainly statistically significant. These results can be validated visually as well, by simply looking at the predicted growth and comparing that to where growth occurred based on newer imagery. While this is less statistically rigorous, the correlation between predicted growth and new man made developments seemes related.

Although this is a novel method of classification and change prediction, it may not be the most accurate. More traditional methods of change prediction such as hotspot analysis may be more appropriate for determining if a specific area is likely to change in the future. Although part of this project was working within the data bounds on purpose, would a prediction using all of the data available within the imagery lead to similar results? To test this, a very basic hotspot analysis could be done on the change detection rasters to highlight where the concentration of Undeveloped to Developed pixels was increased. However this may be a moot point as the value of the CNN, and the reason the prediction was done this way to begin with was the expandability. Even though the “change” or “no change" can be directly generated from the images, that is only after classification. The point of the CNN model is to be saved and used to classify and predict growth in images on the fly, rather than having to be retained everytime it encounters a new image of Phoenix. To train the model takes a lot more thoughtful input via multiple runs of a classification algorithm over multiple images that have been divided in a particular way, this is much less practical than having a trained model that already has those aspects factored into it.

Discussion and Conclusion

There is certainly a correlation between location of growth and the prediction that the CNN model gives. However on the wider scale of the overarching project, there is still much to be done to get the project working as intended as a stand alone application. The memory issues mentioned earlier revolved around reserving enough computing resources to effectively train the model on the full scale of the data. With limited computing resources the model was restricted to running on only a portion of the data. The largest piece of functionality not present in this iteration of the project which will hopefully be included later is the ability to take each of the cells and stitch them back together after the prediction has been made. While this can be done manually, it should be included within the scope of the application. Purposefully limiting the input data to just images restricts the amount of analysis that can be performed with the given data, however complexity comes at the price of usability. The overall goal of this project is to be adapted into a piece of software that only requires images to run so sticking to those parameters is an important requirement. While more data could certainly be added to increase accuracy, the focus of the project was to work within limited data bounds.

Another issue that needs to be addressed is the access to the USGS EarthExplorer API, as stated within the Data section, the API access request made for this project was denied. The code is included within the notebook found in the repo for this project, but it was never able to be tested as I was never provided an API key by USGS.

With that being said, the progress made within the scope of this project, especially in regards to the method and process of preparing and training the model accurately, was not insignificant.

References:

City Growth Detection:

Hussein D. Mohammed , Muhsin A. Ali. Monitoring and Prediction of Urban Growth Using GIS Techniques: A Case Study of Dohuk City Kurdistan Region of Iraq. International Journal of Scientific & Engineering Research, Volume 5, Issue 1, January 2014 ISSN 2229-5518

Arid Classification:

Perez-Aguilar LY, Plata-Rocha W, Monjardin-Armenta SA, Franco-Ochoa C, Zambrano-Medina YG. The Identification and Classification of Arid Zones through Multicriteria Evaluation and Geographic Information Systems—Case Study: Arid Regions of Northwest Mexico. *ISPRS International Journal of Geo-Information*. 2021; 10(11):720. https://doi.org/10.3390/ijgi10110720

Gaur, M.K.; Squires, V.R. Geographic Extent and Characteristics of the World's Arid Zones and Their Peoples. In *Climate Variability Impacts on Land Use and Livelihoods in Drylands*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 3–20.

Pande, C.B., Moharir, K.N., Khadri, S.F.R. *et al.* Study of land use classification in an arid region using multispectral satellite images. *Appl Water Sci* 8, 123 (2018). https://doi.org/10.1007/s13201-018-0764-0