GEOG596B
Augustus Wright
Penn State University, MGIS
Capstone Results

Authoritative Geospatial Data Validation via Machine Learning Algorithms

Table of Contents

I. ABSTRACT 2
II. INTRODUCTION 2
III. LITERATURE REVIEW 3
IV. RESEARCH 4
RESEARCH QUESTIONS: 5
V. INTEGRATING ML WITH GEE 5
VI. THE ANALYSIS 7
METHOD 1: 9
METHOD 2: 10
DATA PREP: 11
METHOD 3: 12
VII. THE RESULTS 13
VIII. PRESENTATION VENUE 14
IX. COMMUNITY FEEDBACK 16
X. REFERENCES 18
WORKS CITED 20

I. Abstract

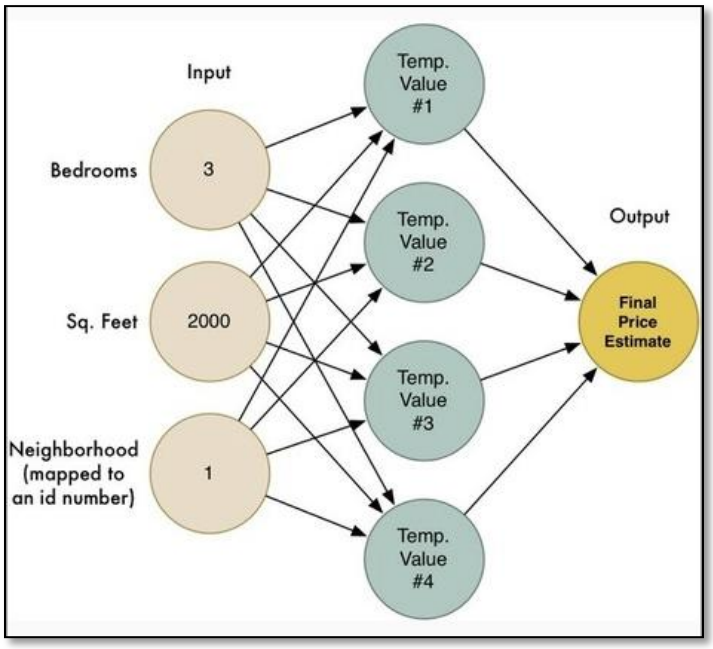
This analysis was intended to explore a modern method of autonomously extracting authoritative vector data by incorporating machine-learning (ML) algorithms into a commonplace GIS extraction Environment (GEE). The purpose of this research is to advance authoritative geospatial data production methodologies at government Geospatial Planning Cells, which primarily create data by heads-up digitizing. From August 2015 to July 2016 I was fortunate enough to take part in a government exchange program called Train with Industry (TWI). The assignment was at Harris Geospatial with the team that develops ENVI (The Environment for Visualizing Imagery) software. This is where I was first introduced to ML applications. The more I learned about this emerging capability in the realm of remote sensing, it became apparent that ML could significantly improve efficiency in authoritative data creation if applied effectively.

II. Introduction

The overarching goal of the project was to determine how to implement ML algorithms in a GEE, how to train an artificial neural network to classify a remotely sensed image, how to autonomously extract specific features from the classified raster, then determine if results are repeatable on a different image. It's important to note that this was an upfront labor-intensive undertaking so scope management was paramount to successful tests. Training the neural network required numerous test iterations that eventually lead to backpropagation. (Programmer, 2016) Backpropagation is shortened for "backward propagation of errors". This is found to be the most common method of training artificial neural networks. It couples with select optimizers and requires that the activation function used by the artificial neurons become able to discern differences between features. (Programmer, 2016) For purposes of confirming or denying plausibility, I only attempted to extract tree canopies and drainage features in these initial trials. All other features are therefore beyond the scope of this project for now. This determination was made after careful review of the chosen literary works. Relationships between the publications, major themes, gaps, and disagreements were all contributing factors to the decision to limit the scope until it is proven that the aforementioned two basic feature sets could be extracted through automated ML approaches from multi-spectral ortho-imagery. (M. Kanevski, 2008)

III. Literature Review

A reoccurring nuance connecting several major concepts was apparent throughout the literary works and proceedings. The relationships involve the development and usage of artificial neural networks, linear regression, geostatistics, variography, binary classifications and a series of model training iterations for the aforementioned neural network. A neural network is essentially a computer-coded model of the human brain.



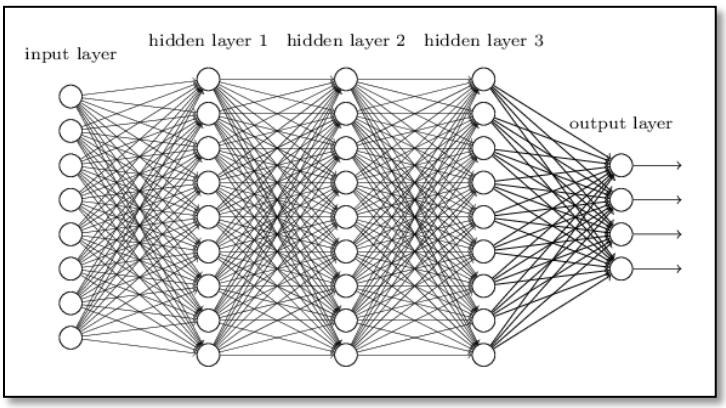
Example Neural Network (Pintado, 2016)

(Programmer, 2016) In the simplest form, artificial neural networks consist of an input tier, one or more hidden computational layers and an output, as depicted in the Example figure to the right.

Neural networks, also known as artificial intelligence (ai), are not anywhere near stages of sophistication depicted in television science fiction, although they are complex enough for tasks such as image/data classification. (M. Kanevski, 2008) There are several Free and Open Source (FOSS) R and Python site packages that lay the groundwork such as TensorFlow, OpenAI, H20ai and many others (Data Mining, Analytics, Big Data, and Data Science, 2016). A comprehensive list of these and more can be explored at http://www.kdnuggets.com/. These modules aid in developing and training neural networks and significantly reduce the need for new development. This is important because neural networks can be simplistic in nature, or very complicated as depicted in the diagram below by Michael Nielsen. This is one determining factor for selecting a preexisting package that already suits this use case for purposes of managing overall scope. In Michael Nielsen's diagram, it's apparent that neural networks consist of an input layer for data entry, one or more hidden computational layers, and lastly an output decisive classifier tier, which is where the machine makes a determination (Nielsen, 2016). Of the many existing packages, I began trials using H2Oai through Python with a linkage to ESRI's ArcGIS via Anaconda. Anaconda is one of the most prominent data science FOSS platforms, which is driven by Python.

IV. Research

Throughout my research, I've identified gaps and disagreements. The biggest gap was how to relate the neural network to GIS data. The research identified Python as being the most feasible solution to bridge this gap. Disagreements primarily concern training methods and which statistical model, ie. Kriging, Sigmoid activation, Softmax activation, or Gaussian is best for taking into account phenomena such as autocorrelation (Ostermann, 2015). The best-fitted statistical model would be determined during trials based on classification results.



Example Neural Network (Nielsen, 2016)

During the research phase, I suspected this aspect of the study would be subjective and primarily dependent upon aspects of the data, and collection platform. The greatest risk to the project was a deep learning problem called overfitting. Overfitting is simply when the ML model learns the data instead of the task it's being trained to automate. (Nitesh V. Chawla, 2002) This problem is best addressed through trial, error, training, and testing.

The research assumption was that this would be a challenging yet rewarding task. In the early stages of research the initial understanding was that in order to successfully automate authoritative feature extraction, a combination of skills in geostatistics, object-oriented programming, and data science would come into play. The anticipated payoff would be a less subjective methodology that enhances a mundane, yet critical workflow. For starters, the approach was to start with two features, tree canopies, and drainage features, for proof of concept. These are 2 of 6 Thematic Layers, (Obstacles, Surface Configuration/Slope, Soils/Surface Material, **Surface Drainage**, Transportation, **Vegetation**), that provide a basic foundation for military analysis, planning and map production (Departments and Agencies of Department of Defense, 1996).

Research Questions:

1. How to best implement ML in a GIS Extraction Environment (GEE)?

2. What ML algorithm(s) best support the desired results?

3. Are the results repeatable on an adjacent sheet?

V. Integrating ML with GEE

After further research and hands-on trials, I narrowed down the method to H2Oai using Python by way of Anaconda while leveraging ESRI modules by December 2016 (USGS, 2014). This was mainly due to several failed attempts to initialize Tensorflow and Openai Gym through ArcGIS Desktop's Python window. Although H2Oai would not initialize directly in ArcGIS, it would launch in a Graphic User Interface (GUI) headless capacity using the Python 2.7.10 executable and/or command window, which both install with ArcGIS Desktop. Following this procedure, H2O did not pose any significant interoperability issues as long as the H2Oai local installation's version is synchronized with the same version of the H2O Python package. Lastly, it was important to ensure the Python numpy version complies with H2Oai (H2oai, 2016).

Several weeks into the analysis it was determined that H2Oai's syntax was more of a learning curve than anticipated. Acquiring H2Oai tutorials and detailed Python documentation began to filch time from the overall project so the decision was made to forego further testing using H2Oai. Although Google's Tensorflow ML library requires a 64bit environment, contradictory to ESRI's 32bit ArcGIS Desktop, it became the most viable option due to ease of use since the syntax is by and large pure Python.

Using Tensorflow required a means of parsing these data out from ArcGIS to a format that could be read by the 64bit neural network. Going forward training, testing, and analysis would be performed on four band National Agriculture Imagery Program (NAIP) aerial imagery acquired at a resolution of 1-meter ground sample distance (GSD) (USGS, 2014).

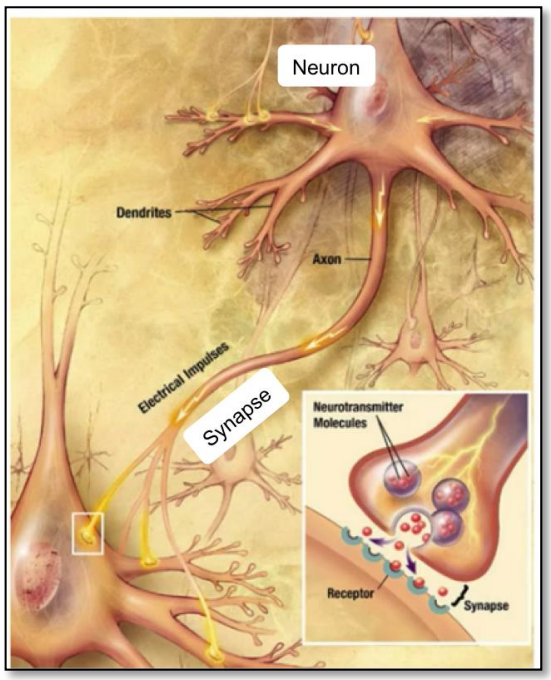
Now that the environment has been set I used ESRI's arcpy site package ~ arcpy.RasterToNumPyArray function to generate digital arrays of raster pixel values from the NAIP ortho-rectified image. The generated arrays were then saved to CSV, which is a comma separated values file, which allows data to be saved in a tabular organized structure.

Using this data preparation method, a refined series of CVS's consisting of the spectral indices served as input training and test independent variables for purposes of training the neural network to recognize pixel values that are likely vegetation or water, based on the values themselves as well as adjacent pixels. Each of the following indices enables the ML algorithm to consider principle elements of image interpretation such as location, size, shape, shadow, tone/color, texture, pattern, height/depth and site/situation/association, and even autocorrelation (Wynne, 2011):

- Near Infrared Mean Segmentation
- Red Mean Segmentation
- Green Mean Segmentation
- Normalized Difference Vegetation Index (NDVI)
- Normalized Difference Wetness Index (NDWI)
- Perceived Luminance Index
- Digital Elevation Model (DEM)
- Normalized Digital Surface Model (nDSM)

In this application of ML in a GEE, the independent variables serve the artificial intelligence in the same way that the five basic senses of sight, hearing, touching, smelling and tasting serve human beings in our ability to discern the gist of an image scene. Historic academic analyses of scene discernment such as Aude Oliva's article, "Gist of the Scene", have shown that human observers can recognize and mentally classify a real-world scene in a single glimpse. The phrase gist of a scene simply means an observer can rapidly identify a variety of perceptual and semantic information given just a momentary glance at a multipart real-world image (Oliva, 2011).

Theoretically, this is made possible through memories and training stored within the organic neural network that makes up the human brain (L. Nadel, 2000). A microscopic Example figure is presented to the right.



Human Neural Network (Eremenk, 2016)

The five human senses serve as the catalyst to an efficient process, which begins with seeing. Next, the visual system passes data along organic synapses to deep neurons where humans form a spatial representation of the outside world that is rich enough to grasp the meaning of the scene (Oliva, 2011). A series of both inductive and deductive logic processes then facilitate recognition of a few objects and other noticeable information in the image, to accelerate object detection (Oliva, 2011). This enables humans to extract by hand albeit subjective from person to person due to several factors such as education, training, experience and even steadiness of the hand.

Using the aforementioned spectral indices maximize the use of both pixel-based classifications as well as principles of Object Based Imagery Analysis (OBIA) to enable an artificial intelligence to extract as humans would. This approach explicitly takes into account elements of image interpretation such as spectral profile, autocorrelation phenomena, being that things closer together tend to be more alike than things further away, as well as other elements like geometry, symmetry, elevation, and brightness (Yuanrong He, 2016). Furthermore, the proposed approach augments human shortcomings such as unsteady hands, fatigue, bias, and restriction to only five known senses, to name a few restraints.

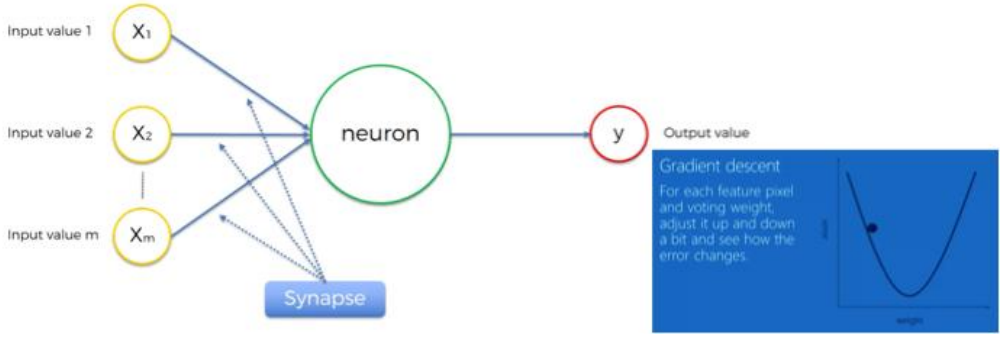
The intent is that the neural network would be provisioned illimitable spectral indices to be used in the same manner that humans use our five senses so that it can classify pixels accordingly into a new binary array for conversion into map data (Eremenk, 2016). The machine would programmatically determine for itself, which senses/ independent variables best enabled it to determine the gist of a scene, just as humans seamlessly only employ the exact number of senses necessary to distinguish features.

VI. The Analysis

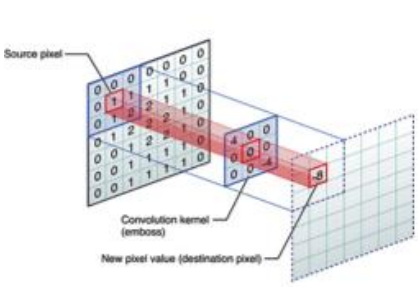
Feature Extraction and several other military automation challenges are best generalized as simple classification problems. Image classification refers to the task of specifying information or feature classes from a video, or multiband raster image (ESRI, 2017). It depends on the interaction between an interpreter and what is being viewed. In traditional workflows the interpreter is human, however, this research aims to validate the feasibility of implementing an artificial interpreter in a GEE. The goal of this analysis is to train an artificial neural network to classify the features in question within 75-80% accuracy, meaning the machine closely mimics the choices made by a human. Human predictions are supplied to the machine in the form of a training dataset traditionally known as a binary terrain categorization TERCAT. The binary TERCAT is the dependent variable that the machine uses to learn what is expected of it, or in other words what ‘right’ looks like.

Employing the scientific method, three ML algorithms were tested to determine which would garner the best results. Those tested were:

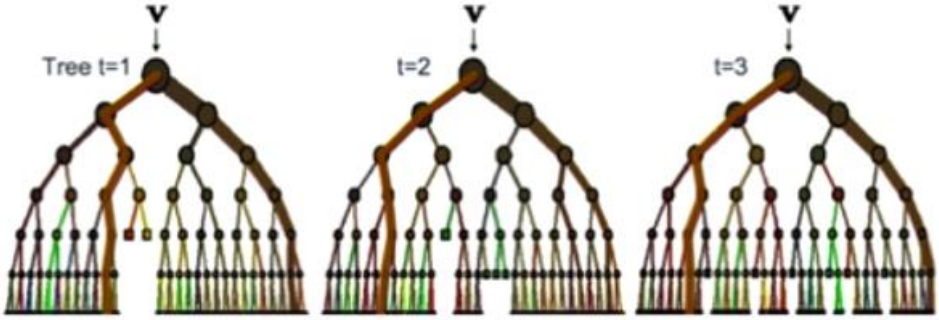
- Artificial Neural Network (ANN) (Eremenk, 2016)



- Convolutional Neural Network (CNN) (Eremenk, 2016)



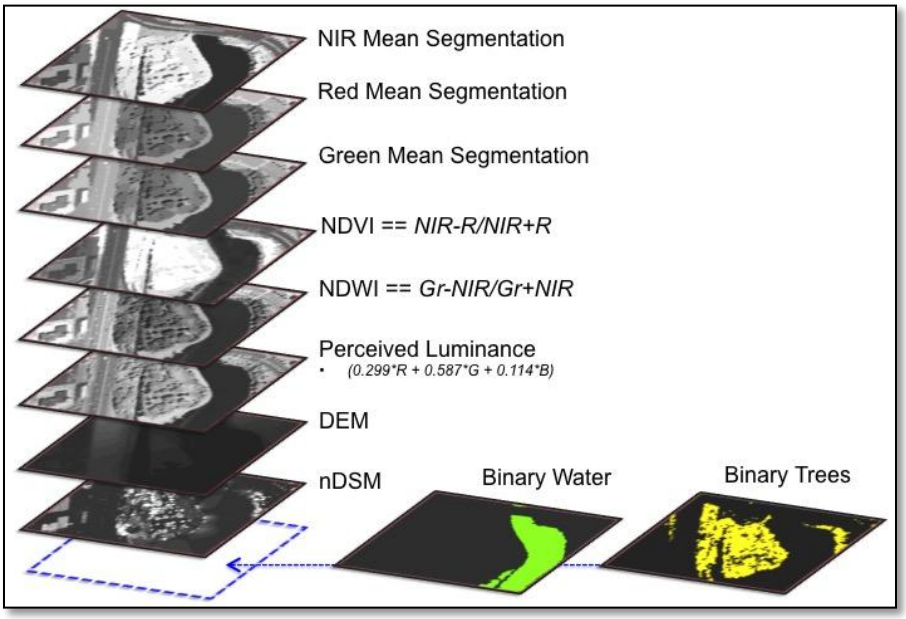
- Sklearn Ensemble (SciPy developers, Scipy.org, 2017)



Each of these methods is premised on the use of independent variables passed through artificial neurons, but in slightly different ways. Backward propagation of errors is also a universal nuance of each.

Method 1:

During training, the ANN programmatically peers down through each of the interdependent variables presented in the Reference figure below, then makes a prediction. The ANN then compares its prediction to the correct human prediction presented in the binary TERCAT. The ANN then iteratively goes back to its artificial synapses and adjusts weights on each sense/independent variable until it is able to come as close as possible to make the same decision that its human trainer would have made. This backward propagation of error process is called Stochastic Gradient Descent (Eremenk, 2016). This process is repeated in thousands of iterations until the machine is able to accurately discern the feature in question.



Spectral Indices/ Dependent Variables (Wynne, 2011)

In this method, I was able to attain 98% precision but less than 20% accuracy. This is determined to be due to a lack of training data. The ANN was wrong, but wrong in the exact same way each time. This approach still has merit but requires further testing on a sufficient workstation than my MacBook Pro.

Training the artificial neural networks was an extremely labor-intensive undertaking in terms of data preparation and processing time. Much of this is to do with my computing environment that consists of a MacBook Pro running Windows 10 through a Parallels Desktop Virtual Machine. Performance will significantly improve, provided an environment configured for pooling and multi-threading.

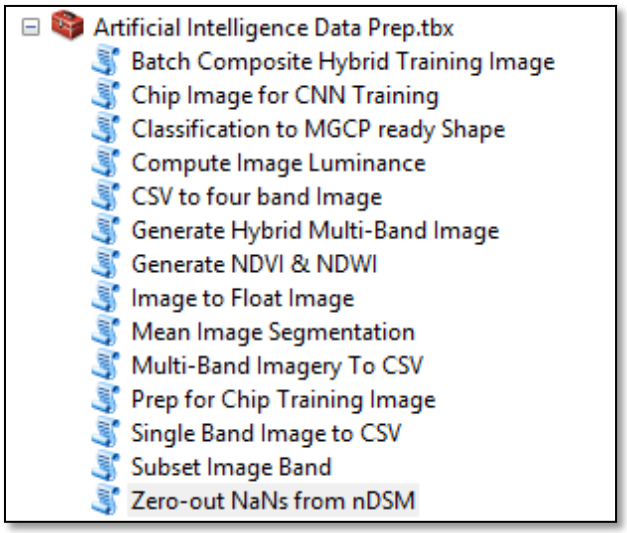
Method 2:

CNN's are the engines behind image classification technology such as face recognition and other image classification achievements. This technology is also commonly referred to as Computer Vision (Eremenk, 2016). In CNN's a pooled sample of features are repeatedly applied as filtered iterations over an image. In other words, it is the act of comparing spectra and spatial patterns then making a classification decision, trying every possible match. The difficulty and beauty in this are that computers are far more literal than humans. This creates difficulty because explicitness is a necessity, otherwise, the prediction will not be as expected. The beauty of it is that once the labor-intensive training is done, the machine will be able to discern classification to an explicitly set standard much faster and more thoroughly than human extractors without any personal subjectivity aside from that which might exist in the training data or code.

During training, the CNN is provided thousands of training images in a file structure. The CNN then iterates through each of the images and learns elements of each image, which constitute a 1/TRUE classification. The CNN is then presented independent images for validation purposes. During validation, the CNN is tested on how well it classifies features to be 1/TRUE matches or 0/FALSE non-matches (Eremenk, 2016). This approach to image classification is useful for human-aided classification. In the future, I intend to explore ways to enhance mouse cursor functionality via C# with computer vision tool tips that leverage CNN's.

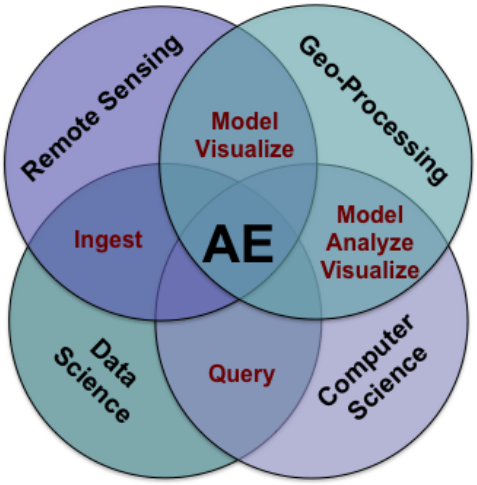
Using CNN's I was able to attain 85% precision but less than 17% accuracy. This is also determined to be due to insufficient of training data. Like the ANN method, the results of the CNN approach are also non-conclusive at this time, although the approach still holds merit for future testing using a sufficient workstation.

Data Prep:
After the first two trials of testing, it became apparent that data wrangling is one of the greatest challenges associated with ML in GEE's. Data wrangling is defined, as is the process of cleaning and merging uncontrolled and complex data sets to ease accessibility and analysis (Datawatch Corporation, 2017). The process of data wrangling commonly includes manually converting and mapping data from one raw format into another to facilitate expedient ingestion and arrangement of these data.



Tools for Redundant Work (ESRI, 2017)

In response to this problem, I began developing the ESRI Script Tools shown in the reference figure to the right for tasks I found myself repeating over and over again during data preparation and neural network training. Going forward, I intend to document these tools more comprehensively and work with associates at ESRI and ENVI to condense any unforeseen inefficiencies within the current code. From there I'll work with both organizations and The Distributed Common Ground System-Army (DCGS-A) to implement the tools onto the DCGS-A baseline for consumption. DCGS-A is the Army's intelligence, surveillance, and reconnaissance (ISR) enterprise for the tasking of sensors, analysis and processing of data, exploitation of data, and dissemination of intelligence (TPED) across military echelons.



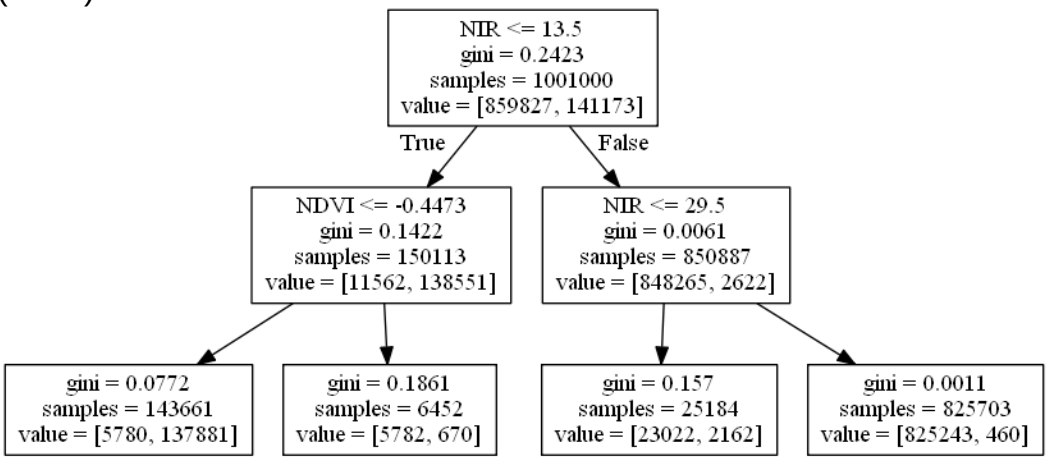
Foundations of Automated Feature Extraction

The Artificial Intelligence Data Prep Toolbox was developed based on Data Scientist Charles Kelly's **IMQAV** acronym (Kelly, 2016). The acronym stands for **Ingest**, **Model**, **Query**, **Analyze** and **Visualize**. I developed the following Venn diagram to depict how these elements must come together in order to achieve Automated Extraction (AE):

Method 3:

In the final test, a method known as ensembling was implemented using the Sklearn Scientific Python site package (SciPy developers, Scipy.org, 2017). Ensembling is one of the most widely accepted, and most powerful machine learning algorithms. It is a type of collective machine learning algorithm referred to as Bootstrap Aggregation, which is also recognized as bagging. In this method up to thousands of decision trees with varying depths are aggregated to enable the machine to string together complex logic to match training criteria in the way that is most efficient for the computer (Brownlee, 2016).

For instance, the Example below indicates that given a maximum network depth of 2, and 8 independent variables as previously mentioned; the main senses/independent variables required for the artificial intelligence to make the same classification decision as the human trainer were Near Infrared (NIR) Mean Segmentation, and Normalized Difference Vegetative Index (NDVI).



Graphic 2 Deep Ensemble for Tree Canopy Prediction (SciPy developers, Scipy.org, 2017)

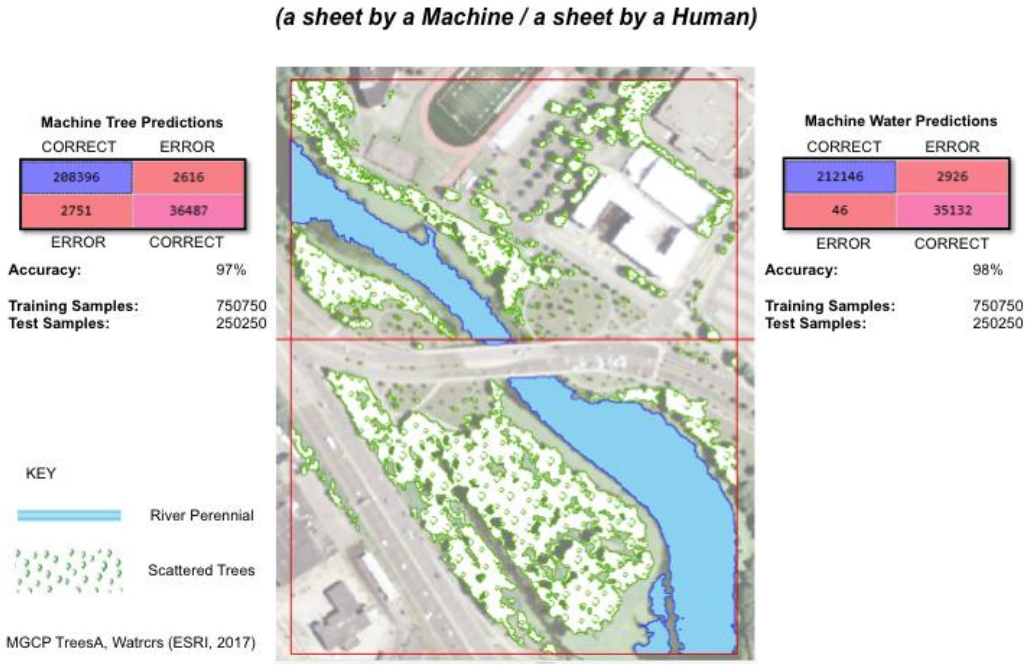
This method proved to be highly efficient because there is less concern about discrete networks overfitting the training data. This enables the neural network to dynamically grow as deep as it needs during training in order to mimic human classification decisions as accurately as possible. This results in high variance and low bias. This technique is commonly referred to as bagging (Brownlee, 2016). Bagging proved to be a highly effective means of incorporating ML in a GEE because it required much less training data than ANN's and CNN's.

The ensemble neural network training was given a depth of 1500 estimators. A kfold cross validation method was implemented to maximize use of the available training data, which consisted of $1{,}001{,}000^8$ spectra, accounting for 8 distinct indices. This means the data was split into X_train, y_train, X_test, and y_test datasets. This is common practice when performing supervised machine learning experiments. This approach subset out part of the available data as a test set X_test, y_test to rule out overfitting (Eremenk, 2016).

VII. The Results

Method 3 proved to be the most effective. The below results are a sheet extracted by the Ensemble algorithm compared to an adjacent sheet extracted by a Human. The machine was able to mimic human discernments of tree canopies at 97% accuracy and surface water at 98% accuracy according to the confusion matrices. This accuracy report is based on the training TERCAT kfold that consists of 750750 training spectra and 250250 test spectra. The neural networks use the kfold to count how often it made choices contradictory to the way the human trainer decided. ERROR labels indicate this.

In this proof of concept, the output predictions were ingested into the Multinational Geospatial Co-production Program (MGCP) feature extraction schema. Ingestion into MGCP schema was made possible by automating F_CODE attribution. The F_CODE is an ordinal field heading used to assign features to specific categories. Additional attribution can also be automated.



Research Results with Accuracy Confusion Matrices (ESRI, 2017)

The MGCP schema translates seamlessly to the National Geospatial-Intelligence Agency's (NGA) Topographic Data Store (TDS) as well as the Army's Ground-Warfighter Geospatial Data Model (GGDM). The MGCP was used for this proof of concept because it requires the least possible amount of attribution required to perform rapid analysis, planning and map production. The source code can be manipulated to ingest directly into TDS or GGDM.

VIII. Presentation Venue

The results of this experiment were presented at the 5th Annual Army Geospatial Planning Cell (GPC) Working Group, which met on 9-11 May 2017 at Fort Leonard Wood, MO. This venue was established to facilitate information exchange, professional development, and analysis on topics that affect all globally positioned GPCs and Geospatial Engineers in the United States Army. Several key organizations attended including the seven GPCs, Army Geospatial Center (AGC), Maneuver Support Center of Excellence, NGA, Army Engineer School, Headquarters Department of the Army G2, Intelligence Center of Excellence, Office of the Chief of Engineers, Program Management Office DCGS-A, ESRI, Harris IT, Penn State University and the Army Training and Doctrine Capabilities Manager (TCM) Geospatial.

From July 2012 to July 2015 I served as Direct Support GEOINT Officer in Charge at the 60th GPC in Wiesbaden Germany. There I took part in developing and improving geospatial production in Europe through efforts that enabled the 60th GPC to support three Contingency Operations Commands (EUCOM, AFRICOM, and SOCOM), two Army Service Component Commands (USAREUR and USARAF), and two Special Operations Commands (SOCAF and SOCEUR) with combined data collection, analysis, production, dissemination of products and web-enabled geospatial services. During this assignment I became very familiar with the intricacies of the Army GPC mission that is basically to generate authoritative vector data, hence is why this venue was selected to share the research results.

During the presentation the attendees were asked to answer the following six questions:

Please specify your main reason for attending this presentation:

Which aspects were you mostly interested in?

What is the most beneficial aspect of the project?

What is the least beneficial aspect of the project?

What impact do you think the project will have on Army Geospatial, if any?

List any questions or concerns you have that were not answered or addressed:

Email: augustus.wright.mil@mail.mil

IX. Community Feedback

The responses indicate a universal nuance that suggests most attendees sat in on the presentation because they viewed it as an opportunity to gain insight on an emerging technology that aligns directly with the future trajectory of geospatial engineering and GEOINT. Many found the 97-98% accuracy to be the most compelling aspect. Based on the feedback, the accuracy is also viewed as being the most beneficial aspect of augmenting the current workflow with ML algorithms. This audience, by and large, sees this as an opportunity to significantly reduce human error when generating authoritative vector data from remotely sensed imagery.

Despite the enthusiasm evident from responses to the first three questions, the audience found the technical aspects of ML to be quite difficult to understand. The sentiment was that a 30-minute presentation did not provide the level of detail required to foster a more comprehensive understanding.

Nonetheless, the audience thinks the technology will positively impact U.S. Army GPCs and NGA significantly by saving time, and dollars while bolstering standardization. During the research phase of the project, empirical data surrounding the cost associated with vector generation was traced back to two analyses from 1979 and more recently 2010.

In 1979 George Hanuschak performed an observed analysis to record the cost of digitizing crop areas by hand from LANDSAT imagery. The results of his analysis found that on average a 2.59 Square Kilometer map sheet took roughly 1 hour to digitize once the environment was set for extraction. The project comprised 298 map sheets and cost the United States Department of Agriculture $300,000 to complete over an 11 month period. Computers and software have advanced since then, but methods used, for the most part, are the same until now. The name of this report is “Obtaining timely crop area estimates using ground-gathered and LANDSAT data” for those that wish to read further into it (Hanuschak, 1979).

More recently, in 2010 a comrade of mine, CW4 Scott Hashagen, performed a Lean Six Sigma analysis of an Army Geospatial Planning Cell’s annual production cycle. His study found that the variation of imagery and data extracted by each analyst made it difficult for the data Steward to perform quality assurance and control measures. Through this study the unit determined that up to 2 man-months were lost per sheet as a result of poor ergonomics and effort being duplicated at step 3, topology checks, edge matching and conflation, causing a backlog for the data Steward. Inexperience, training shortfalls, subjectivity and the need to recreate blatantly bad data from scratch contributed to this inefficiency. It all equates to even

greater labor percentage costs when salaries are taken into account (Hashagen, 2010).

CW4 Scott Hashagen and his Officer In Charge LTC Craig Guth were able to decrease the production cycle down from 2 man months per sheet to 1 man month by improving ergonomics. Updating the SOP, conducting a training stand-down, and incorporating digitizing monitors and digital pens achieved this cost reduction (Hashagen, 2010). Despite the significant improvement, the measures taken do not maximize the spectral information available from today's sensor platforms, do not maximize computing power and nor do they eliminate human error.

The Army Geospatial Community recognizes ML algorithms as having the merit to reduce the authoritative data production cycle from a man month per sheet to potentially man hours per sheet. COL John Connor, the TCM-Geospatial, sees this as an opportunity to gain 1/4th of a Soldier at each GPC that never has to sleep, take breaks, or eat. ML algorithms can work around the clock to fill operational data gaps. Furthermore, artificial intelligence augmentations possess insights and increased ability to make explicit ground truth interpretations.

X. References

- Major Works
  - Machine Learning for the Detection of Oil Spills in Satellite Radar Images: MIROSLAV KUBAT* (MIROSLAV KUBAT, 1998)
    - Applicable Responses
      - 1a. Synthetic Minority Over-sampling Technique: Nitesh V. Chawla (Nitesh V. Chawla, 2002)
      - 1b. Robust Classification for Imprecise Environments: FOSTER PROVOST (PROVOST, 2001) (D. G. Brown †*, 2000)
  - Ma Modeling the relationships between landuse and land cover on private lands in the Upper Midwest, USA: D. G. Brown†*, B. C. Pijanowski‡ and J. D. Duh† (D. G. Brown †*, 2000)
    - Applicable Responses
      - 2a. Machine Learning Algorithms for GeoSpatial Data. Applications and Software Tools: M. Kanevski, A. Pozdnoukhov, V. Timonin (M. Kanevski, 2008)
      - 2b. Hybrid geo-information processing:
      - Crowdsourced supervision of geospatial machine learning tasks: Frank O. Ostermann (Ostermann, 2015)
- Convention Preceding's
  - ENVI Analytics Symposium 2016 (MEGA ML Algorithm) (Harris Visualization ENVI, 2016)
  - Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications (ACM DL, 2011)
- Books:

- Deep Learning Fundamentals in Python: Lazy Programmer
  Lazy Programmer: (Programmer, 2016)

- Other Sources

  - Introduction to Data Science: Lynda.com (Poulson, 2015)

Works Cited

ACM DL. (2011). *Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications.* Washington, DC: ACM DL.

Brownlee, J. (2016, April 22). Bagging and Random Forest Ensemble Algorithms for Machine Learning. *Machine Learning Algorithms*.

D. G. Brown †*, B. C. (2000). *Modeling the relationships between land use and land cover on private lands in the Upper Midwest, USA.* Journal of Environmental Management. Midwest, USA: Academic Press.

Data Mining, Analytics, Big Data, and Data Science. (2016, November). Data Mining, Analytics, Big Data, and Data Science. USA.

Datawatch Corporation. (2017). What is Data Wrangling? *What is Data Wrangling?* USA.

Departments and Agencies of Department of Defense. (1996). *Performance Specification Vector Product Interim Terrain Data (VITD).* Washington, DC.

Eremenk, K. (2016). Deep Learning A-Z: Hands-On Artificial Neural Networks. USA.

ESRI. (2017). ArcGIS Help. Redlands, CA, USA.

H2oai. (2016, November 14). H2O, Sparkling Water, and Steam Documentation. USA.

Hanuschak, G. (1979). Obtaining timely crop area estimates using ground-gathered and LANDSAT data. *United States Department of Agriculture*.

Harris Visualization ENVI. (2016). *ENVI Analytics Symposium Proceedings on MEGA.* Boulder, CO: Harris Corporation.

Hashagen, C. S. (2010). *Lean Six Sigma analysis of an Army Geospatial Planning Cell's annual data production.* United States Army. United States Army.

Kelly, C. (2016). Data Science Foundations: Data Mining. *Data Science Foundations: Data Mining*. USA.

L. Nadel, A. S. (2000, September). Multiple trace theory of human memory: Computational, neuroimaging, and neuropsychological results. *Hippocampus* .

M. Kanevski, A. P. (2008). *Machine Learning Algorithms for GeoSpatial Data. Applications and Software Tools.* Institute of Geomatics and Analysis of Risk (IGAR),

Faculty of Geosciences and Environment, University of Lausanne. Lausanne, Switzerland: University of Lausanne.

MIROSLAV KUBAT, R. C. (1998). *Machine Learning for the Detection of Oil Spills in Satellite Radar Images.* School of Information Technology and Engineering, University of Ottawa. Boston: Kluwer Academic Publishers, Boston.

Nielsen, M. (2016, 01 01). Neural Networks and Deep Learning. *Determination Press*. USA.

Nitesh V. Chawla, K. W. (2002). *SMOTE: Synthetic Minority Over-sampling Technique.* Journal of Artificial Intelligence Research. Tampa, FL: Journal of Artificial Intelligence Research.

Oliva, A. (2011). The gist of the Scene. In A. Oliva, *Neurobiology of Attention.* Journal of Neuroscience.

Ostermann, F. O. (2015). *Hybrid geo-information processing: Crowdsourced supervision of geospatial machine learning tasks.* University of Twente . AE Enschede, The Netherlands: University of Twente.

Pintado, J. H. (2016, October 01). Errors Are Imminent. *Computer Science, Programming, Maths and Big Data*. Somewhere, Over the Rainbow, USA.

Poulson, B. (2015, 1 1). Introduction to Data Science. State College, PA, USA.

Programmer, L. (2016). *Deep Learning Fundamentals in Python.* LazyProgrammer.

PROVOST, F. (2001). *Robust Classification for Imprecise Environments.* The Netherlands: Kluwer Academic Publishers.

SciPy developers, Scipy.org. (2017). Scipy Site Package Documentation. USA.

USGS. (2014, JAN). Using Anaconda modules from the ESRI Python environment (All Users). Rolla, MO, USA.

Wynne, J. B. (2011). *Introduction to Remote Sensing Fith Edition.* New York, NY, USA: Guilford Publications.

Yuanrong He, X. Z. (2016). Object-Based Distinction between Building Shadow and Water in High-Resolution Imagery Using Fuzzy-Rule Classification and Artificial Bee Colony Optimization. *Journal of Sensors*, 10.