United States Department of Agriculture Forest Service

California Urban School Yard Mapping Geospatial Analysis Methodology and Findings

Report by Eric Romero

July 2024

PhD Candidate, the University of California, Berkeley

Department of Environmental Science, Policy, and Management

Table of Acronyms

United States Department of Agriculture	USDA
National Aerial Imagery Program	NAIP
Machine Learning	ML
Random Forest	RF
Google Earth Engine	GEE
Environmental Systems Research Institute	ESRI
Geographic Information System	GIS
Contrast Limited Adaptive Histogram Equalization	CLAHE
Simple Non-iterative Clustering Algorithm	SNIC
Near Infrared Reflectance of Vegetation	NIRv
Normalized Difference Vegetation Index	NDVI
Visible Atmospherically Resistant Index	VARI
Gray Level Co-occurrence Matrix	GLCM
Red, Green, Blue	R, G, B
Near Infrared, Red, Green	N, R, G
Open Street Maps	OSM
Light Detection and Ranging	LiDAR
Synthetic Aperture Radar	SAR
Convolutional Neural Network	CNN

Introduction

This document outlines the steps taken to perform multiple geospatial analysis methods for classifying public elementary school land cover in three major cities in California, USA. The cities of focus for this analysis were (1) Los Angeles, (2) Oakland, (3) Sacramento and the land cover classes were (1) Tree/Shrub, (2) Grass/Pervious, and (3) Impervious. The first component of this analysis involved forming reference image datasets over each California urban center using aerial survey imagery from the USDA NAIP. The second part of this analysis involved collecting training and validation datasets for ML model development and accuracy assessment. The third step involved training, classification, and validation of an RF ML classifier. Additional post-processing steps and analysis involved integration of open-source land cover data for increased granularity in the final land cover dataset, as well as statistical comparison of land cover classes across the three urban centers and previous land cover surveys.

Eligible Elementary Schools

We collected data for schools that met a certain criterion as part of a larger school tree study with the USDA Forest Service and researchers at the University of California, Berkeley, Davis, and Los Angeles.

Eligible schools must be located in one of the following three school districts: Los Angeles Unified, Oakland Unified, and Sacramento City Unified. Further, schools must be public, non-charter elementary schools that either serve grades K-5 or K-6 (i.e., not K-3, 4-6, or any other subset).

Certain schools border or share campus facilities with other schools (termed "co-located"). In this study, we excluded otherwise eligible elementary schools which are co-located with ineligible schools (i.e., those that do not meet the above eligibility criteria, such as middle or high schools) and do not have a clear physical barrier separating the campuses. However, eligible schools that are co-located with ineligible schools, but are separated by a clear physical barrier, were included in this study. In a few instances, two co-located schools which individually meet all initial criteria for eligibility do not appear to be separated by a clear physical barrier. These schools were included in this study and were treated as a single campus (i.e., contained a single combined record), with relevant notes retained in the attribute tables.

NAIP Image Collection

The following image collection steps were conducted for each of the three cities. NAIP imagery was collected from the survey year of 2022 using GEE code editor software. NAIP imagery from each city was acquired on different dates in late spring and early summer. Los Angeles NAIP imagery was acquired on the 4^{th}, 5^{th}, 10^{th}, 11^{th}, and 12^{th} of May 2022. Oakland NAIP imagery was acquired on the 18^{th} and 19^{th} of May 2022. Sacramento NAIP imagery was acquired on the 21^{st} of June 2022. The geometry extent of each urban center was uploaded to GEE as an ESRI shapefile, which was then used as the extent for image mosaicking and masking in an image pre-processing task. All pre-processed imagery was then exported to Google Drive using GEE's batch export tool. From Google Drive, the NAIP imagery could be exported locally for processing in a local Python environment. All exported NAIP imagery included the following image bands at 60 cm spatial resolution: (1) Red (2) Green (3) Blue (4) Near Infrared.

Training and Validation Data Collection

All training and validation data were collected manually on local machines within ArcGIS Pro software. Training data were collected manually via visual inspection of the NAIP imagery to inform a supervised object-based RF image classifier model. Model training data was collected by two collectors – a post-master's intern with the USDA and a doctoral graduate student with UC Berkeley. Because of the object-based nature of this classification scheme, point training data was collected instead of object delineations. A total of 26,684 training points were collected across all three cities in California, with 12,1937 points collected in Los Angeles (~48.5%), 5,277 points collected in Oakland (~19.8%), and 8,470 points collected in Sacramento (~31.7%). Our goal in creating a distribution of training point data was to collect a sample that was representative of all three geographic regions, although there was a slight bias in the number of total points collected in Los Angeles due to the larger number of school campuses relative to the other two cities (Los Angeles – 398 schools, Oakland – 37, Sacramento – 40 schools). All training point data were collected within a 500 m^2 area centered around certain designated elementary school grounds during model training to include school campus and school campus-adjacent land cover. All training data were collected in ArcGIS Pro using the Create Point Features tool, and the training points were collected using the following attribute table schema:

Attribute Table	
FID	LandCover

Table 1. Example of training and validation shapefile attribute table schema

with FID corresponding to the number feature collected and LandCover corresponding to the land cover class label of the training point. The FID and LandCover class consisted of 32 and 8-bit unsigned integers, respectively. The classifications labels for the LandCover class consisted of (1) Tree/Shrub, (2) Grass/Pervious, and (3) Impervious. The number of training points for each class consisted of 11,586 (43.42%) Tree/Shrub class, 5,805 (21.75%) Grass/Pervious class, and 9,293 (34.83%) Impervious class labels. Training point features for the Tree/Shrub class included areas which were deemed by the training data collectors as woody vegetation with apparent shadows adjacent. To make shadows more obvious during training data collection, a histogram equalization was used in conjunction with true color (R, G, B) and false color image (N, R, G) symbology. Training point features for the Grass/Pervious class included areas which were deemed by the training collectors as herbaceous vegetation or other pervious surfaces such as soil. Training point features for the Impervious class included land covers like asphalt, roads, concrete, buildings, sidewalks, cars, playgrounds, and synthetic play surfaces.

Once training data collection was completed, validation polygons were collected using an identical attribute table schema to the training data points. Validation polygons were collected solely by the post-master's intern with the USDA to avoid major discrepancies in validation polygon creation between multiple polygon collectors. All validation polygons were collected in

ArcGIS Pro using the Create Polygon Features tool. For all land cover classes, a total of 1,106 polygons were collected, with 464, 305, and 337 polygons from Los Angeles, Oakland, and Sacramento, respectively. The number of polygons for each class consisted of 617 Tree/Shrub (~55.8%), 231 for Grass/Pervious (~20.9%), and 258 for Impervious (~23.3%). Validation polygon features were collected following the same description as training point features, as described in the previous paragraph.

Random Forest Training, Classification, and Validation

A local machine (Dell Inc. Precision 3650 Tower) was used for the development and implementation of the ML RF image classification algorithm within a local Python programming environment. Due to the nature of the high-resolution NAIP imagery, the image file sizes over the full extent of the cities were extremely large [min: 1 GB, max: 30 GB]. Because of this, batch processing was implemented to reduce the intense memory constraints of this image processing task. A grid polygon shapefile of 500 m^2 areas which intersected the boundaries of a polygon shapefile of all the urban elementary school campuses was created so that the NAIP imagery could be processed in a uniform and consistent manner. One grid shapefile was created for the overarching classification task in each urban center, and subset shapefiles of these grids were extracted for the training process. The classification grid shapefiles for each urban center contained 693, 63, and 65 tiles for Los Angeles, Oakland, and Sacramento, respectively. The training subsets for each grid shapefile included 53, 6, and 8 tiles for Los Angeles, Oakland, and Sacramento, respectively.

Model training involved iterating over each grid tile within the training grid shapefile. Within each grid the corresponding NAIP imagery was masked using the boundary of the grid tile, and individual bands of the NAIP imagery were standardized using a CLAHE from the Scikit-Image open-source Python library (Van Der Walt et al. 2014). CLAHE was used for image standardization to increase image contrast and to account for possible long-tailed distributions which can be common in remotely sensed images. After image standardization, the image was segmented using the SNIC algorithm from the Pysnic open-source Python library (Achant and Susstrunk 2017). The bands used for image segmentation corresponded to a false color band combination of Near Infrared, Red, and Green (NRG) Bands. For each 500m^2 image subset, an initial seed of ~20,000 image segments were seeded using SNIC. After image segments were formed, each object that intersected a training label was iterated over and objects statistics were collected for each band of the image as well as selected spectral indices and one image texture metric. Object statistics include the minimum, maximum, range, mean, median, variance, and standard deviation. The bands, indices, and texture that were included in the training and classification can be found in Table 2.

Training Bands, Indices, and Textures

Red
Green
Blue
Near Infrared
NIRv
VARI
NDVI
GLCM Entropy (Green Band)

Table 2. Image bands, spectral indices, and texture metrics included in image object statistics collection.

Once object statistics had been collected, a corresponding integer label was generated for each object based upon its intersecting training data shapefile point feature. After labeling, the arrays of both object statistics and object labels were passed to a RF classifier object from the Scikit-Learn open-source Python library (Pedregosa et al. 2011). Using an Intel Xeon 10 core 3.70 GHz CPU with parallel processing, image segmentation, object statistics collection, and model training took ~5 hours of computation time.

Once model training was completed, image classification was performed. A similar approach to model training was implemented, with batch processing utilizing a gridded tile shapefile to iterate over the larger NAIP imagery and break it into 500m^2 memory-manageable image subsets. Again, image subsets were standardized using the same CLAHE from Scikit-Image (Van Der Walt et al. 2014). False color (NRG) images were again segmented using SNIC and an initial seed of ~20,000 image segments. With segmentation complete, object statistics were collected to include the minimum, maximum, range, mean, median, variance, and standard deviation of all bands, indices, and textures from Table 2. Once each image subset was segmented and image object statistics collected, the random forest classifier was given each object and its statistics to classify and label as either Tree/Shrub, Grass/Pervious, or Impervious land cover. With each segment classified, segment labels were applied to the original segmented image array to create classified image rasters. The final classified rasters were mosaicked together in ArcGIS Pro software using the 'Mosaic to New Raster' Geoprocessing tool. No-data regions outside of school campuses were masked using a shapefile of the elementary school campus extents which were buffered by 10 meters to include adjacent land cover. After all classification steps were complete, some regions of the classified raster datasets retained a "patchy" appearance and often the boundaries of delineated features appeared rough or jagged. To reduce the jagged and patchy appearance of the final classification an image smoothing technique was applied using a 9x9 pixel moving window majority calculation using ArcGIS Pro's Focal Statistics tool from their Image Analyst toolbox. The final classified raster dataset was also converted to vector data in the form of ESRI shapefiles.

Because this analysis implemented an object-based image classification approach, an area-based accuracy assessment was implemented as described by Congalton and Green (2019). This validation approach was chosen because it weighs the validation of the classification results based upon the relative area that each class contributes to the overall landcover and the actual classes that were applied to the imagery. The results of the validation for all three cities can be found in Tables 3 and 4 in the form of summary accuracy metrics.

City	*Overall Accuracy*	*Kappa Score*
Los Angeles	96.37%	0.902
Oakland	93.52%	0.896
Sacramento	91.33%	0.787
All	93.47%	0.890

Table 3. Accuracy values and kappa scores for each urban center and all cities combined

All Cities				
	User's Accuracy	*Producer's Accuracy*	*F1-Score*	*%Area of Validation Data*
Tree	74.31%	94.03%	0.8301475	10.41%
Grass	98.18%	89.44%	0.93602881	49.87%
Impervious	94.40%	98.39%	0.96356793	39.73%

Table 4. User's/producer's accuracy, F1-score, and percent area of corresponding validation polygon class across all cities

The user's accuracy describes how trustworthy is the label that our model applies, for example if we pick 100 areas classified as trees by the model, ~74 of them are indeed trees. In contrast, the producer's accuracy reflects how good the model is at identifying all instances of a class from our validation dataset. For example, if there are 100 areas that are truly trees in our validation dataset, the model correctly labeled ~94 of them. The F1-score is the harmonic mean of these two metrics.

Integration of Open-Source Land Cover Data

To add granularity to the land cover classification performed using the ML RF model in this analysis, we performed post-processing steps to integrate open-source OSM land cover classification data to our assessment. OSM is a free, editable map of the whole world that is being built by volunteers largely from scratch and released with an open-content license. OSM building polygon shapefile data was downloaded from OSM's North America, California, USA dataset via the Geofabrik download server. The California OSM Geofabrik download service can be accessed through https://download.geofabrik.de/north-america/us/california.html. Once downloaded, OSM building polygons were clipped to 10-meter buffered boundary polygons of the elementary school campuses and merged with the final classified vector dataset as a fourth land cover class (4) Buildings/Structures. Because we merged vectorized versions of the OSM building polygons and our classification results, the location of all features from all datasets was preserved. This is to say that if a tree canopy was classified above a building in our land cover data, the presence of the tree canopy does not supersede or replace the building beneath it. Likewise, the building feature does not replace the tree, both features are able to exist simultaneously in vector space.

Land Cover Class Patterns

The following tables describe summary statistics of land cover class distributions across the school campuses in each urban center. Information regarding land cover class distributions across individual schools can be made available upon request to the authors.

Los Angeles		
LandCover	*Area (m2)*	*%Area*
Tree/Shrub	2,233,127.24	18.91
Grass/Pervious	537,921.55	4.56
Impervious	6,980,885.85	59.13
Buildings/Structures	2,054,829.19	17.40

Oakland		
LandCover	*Area (m2)*	*%Area*
Tree/Shrub	191,910.03	19.81
Grass/Pervious	110,109.34	11.37
Impervious	496,490.22	51.25
Buildings/Structures	170,300.11	17.58

Sacramento		
LandCover	*Area (m2)*	*%Area*
Tree/Shrub	396,741.08	22.76
Grass/Pervious	501,352.93	28.76
Impervious	756,905.12	43.42
Buildings/Structures	88,384.65	5.07

Table 5. Summary land cover statistics for schools across Los Angeles, Oakland, and Sacramento cities.

Comparison to Other Urban Tree Canopy Data

The USDA maintains a tree canopy coverage map over cities across California. This canopy coverage map is produced by EarthDefine through a contract with the USDA. The classification methods used by EarthDefine to produce their tree cover maps are not publicly available. Description of the California urban tree coverage dataset as well as access to a data viewer and options to download are available through https://www.fs.usda.gov/detailfull/r5/communityforests/?cid=fseprd647442 and described further at https://www.fs.usda.gov/detail/r5/communityforests/?cid=fseprd649890. The most recent version of the USDA-EarthDefine tree coverage map was produced using NAIP aerial imagery collected in 2018 throughout California, whereas the land cover classification in this study was produced using more recent 2022 NAIP aerial imagery. Because EarthDefine's tree classification methods are not publicly available, we cannot know if our approach to land cover classification is comparable to theirs. Even so, we have collected some basic summary statistics describing the total tree area in both our estimates and theirs, as well as the percent overlap between the two classifications.

Accuracy Metrics Relative to 2018 EarthDefine Dataset					
Region	*User's Accuracy*	*Producer's Accuracy*	*F1-Score*	*Overall Accuracy*	*Kappa Score*
All Cities	94%	90%	0.92	75.9%	0.96
Los Angeles	94%	91%	0.92	76.1%	0.97
Oakland	97%	90%	0.93	77.3%	0.97
Sacramento	96%	88%	0.92	0.74	0.93

Comparison of Tree Areas and Percent Tree Cover Relative to 2018 EarthDefine				
Region	*Tree Area Our Study (m2)*	*Tree Area EarthDefine (m2)*	*%Tree Cover (This study)*	*%Tree Cover (EarthDefine)*
Los Angeles	2,581,630.20	2,159,209.08	19.14	16.01
Oakland	228,408.84	161,019.72	19.95	14.07
Sacramento	481,262.04	340,615.44	22.58	15.98

Table 6. Comparison tree cover statistics between this study and USDA-EarthDefine city school campus tree cover dataset

Potential Limitations of Image Classification Algorithm

Overall, the accuracy metrics of this classification reflect that the model performed well except for the User's Accuracy for the Tree/Shrub class, which reported the lowest of all three classes at 74.31%. This underperformance may be attributed to two primary culprits throughout the methodology: (1) image segmentation errors and (2) spectral signature misclassification errors.

Image segmentation errors arise because of inhomogeneous image segment formation before classification. The method used in this paper implements a segmentation approach, which aims to break an image into many small pieces called "objects". Typically, image objects try to delineate individual features (e.g., one car, one tree, one building), but doing so is quite difficult due to the heterogeneous nature of remotely sensed imagery. Instead of a classical image segmentation, we elected to use what is referred to as an "over-segmentation" approach, which increases the number of segments used relative to a typical image segmentation. When using an over-segmentation approach and image objects to train an ML classifier, the idea is that large features (e.g., trees, buildings, fields) will be broken into constituent objects, the model will be trained to identify that these constituent objects are simply smaller pieces of a greater whole, and that when classification takes place, the smaller objects will be merged together to form an accurately classified larger object. Therefore, if image segments are formed incorrectly or are heterogeneous, then when classification attempts to merge constituent objects back into whole features, there may be small portions of misclassified features in the final map where one segment consisted of two land cover types. This can typically happen when the interface between two land cover types is similar in appearance, for example the edge of a green tree canopy and the green grass below it is joined into one segment due to their greenness and the two-dimensional nature of the remotely sensed imagery, which makes taller surfaces indistinguishable from lower ones.

Spectral signature misclassification errors may arise because the ML model may interpret the spectral reflectance statistics of a given object to be more like an incorrect land cover class. This again can likely be attributed to the use of over-segmentation before image classification. For example, if a tree canopy is broken up into very small segments and some patches of the canopy are less healthy than others, these smaller and less healthy segments may be misclassified by our model as grass or pervious surfaces. Although the smaller and less-healthy tree segment may still retain the spectral characteristics of vegetation, the statistical distribution of the spectral reflectance values in the segment may be deemed more like grass/pervious land cover based on our model's training. The opposite scenario may also arise, where small segments in an over-segmented field of grass may be classified as tree canopy, resulting in a field with a patchy appearance and filled with small misclassification errors. This can occur for the same reason as before, where parts of the field may be healthier than others, resulting in a greener spectral reflectance in these healthier areas. When over-segmentation takes place, healthy segments are separated from unhealthy ones, and during classification healthy segments are misclassified as tree canopies due to their strong vegetative signal.

One way to overcome the limitations mentioned here could be to incorporate an additional active remotely sensed dataset which describes elevation or physical structure, such as high-resolution (≤ 1-meter) LiDAR or SAR. The combined statistical characteristics of active and passive remote sensing data can help distinguish structure (LiDAR/SAR) of features and potentially their elevation (LiDAR), giving the ML model the last piece of the puzzle to distinguish low-lying herbaceous from taller woody vegetation. Such data fusion approaches with NAIP and LiDAR have been used in previous urban tree canopy mapping (O'Neil-Dunne et al. 2013, 2014).

Possible Next Steps

The procedures and results outlined in this paper are the result of a 2-month graduate research position, but the scope of this research could continue in several ways to augment the work done here, such as (1) classifying and comparing historical NAIP surveys, (2) expanding to additional schools in the district, (3) development of a web application through GEE for public use, and (4) classification accuracy improvement.

Classification of and comparison to historical NAIP surveys will require additional collection of training data from past NAIP imagery and re-training of the model using both current and historical NAIP data. This would entail image calibration across years, down-sampling of current NAIP imagery to match lower spatial resolution of historical NAIP imagery (i.e., resample from high to lower resolution), and retraining of the ML model using both current and historical training data as well as NAIP imagery. Expanding this study to additional schools in each district would only require processing additional portions of the NAIP imagery where those schools reside now that model training is complete. The development of a web application in GEE would require bringing in additional expertise in JavaScript and GEE coding. An intermediate to expert user of JavaScript and GEE could have a public-facing web application operational in an estimated ~2 months with access to the spatial datasets produced in this analysis. A GEE web application is free to use if all of the spatial data displayed can be supported by the memory limitations of their cloud processing.

A final cautionary recommendation for further improvement of machine learning model accuracy would be to use several CNN deep learning models. CNN models are well-known to be one of the state-of-the-art models in image classification tasks, but they come with very steep model training costs. Typically, these models require tens of thousands of training polygons, which are much more tedious to collect compared to using training points and an image object-based model training and classification framework. These models tend to perform very well on binary classification problems, such as individual tree delineation in high resolution remotely sensed imagery (Brandt et al. 2020, Tucker et al. 2023). With sufficient training polygon feature collection, a binary classifier CNN model could be trained for the three land cover types described in this document (i.e., one model for each land cover type). This would require a tremendous effort in training data collection and computation time. Once all three models were trained, they could be used to separately classify the NAIP imagery into three binary classification maps (one binary map for each class). These binary maps could then be merged into a final classification map. This recommendation is cautionary, due to the intense training data collection requirements for these kinds of models, as well as the large amounts of computation time and resources needed to train the models and classify the imagery. As a final note, these models may not produce desired results, and often times in image classification tasks simple models like RF classifiers can produce adequate results relative to their deep learning relatives.

Code Availability

All code created to perform this geospatial processing can be found for free public use via https://github.com/erthromero/CASchoolYardMapping/tree/main.

Acknowledgements

This project was funded by agreement number 23-JV-11272123-013 between J Battles at UC Berkeley and the USDA Forest Service Pacific Southwest Research Station, and supported by the Urban and Community Forestry programs at the USDA Forest Service Region 5 and the California Department of Forestry & Fire Protection. I Dronova advised on methods, S Muir assisted with training and validation data, and N van Doorn, L Roman, A Ossola, and L Bigelow provided feedback on earlier drafts of the mapping products and methods documentation.

References Cited

Achant, R., and S. Susstrunk. 2017. Superpixels and Polygons using Simple Non-Iterative Clustering.

Brandt, M., C. J. Tucker, A. Kariryaa, K. Rasmussen, C. Abel, J. Small, J. Chave, L. V. Rasmussen, P. Hiernaux, A. A. Diouf, L. Kergoat, O. Mertz, C. Igel, F. Gieseke, J. Schöning, S. Li, K. Melocik, J. Meyer, S. Sinno, E. Romero, E. Glennie, A. Montagu, M. Dendoncker, and R. Fensholt. 2020. An unexpectedly large count of trees in the West African Sahara and Sahel. Nature 587:78–82.

O'Neil-Dunne, J., S. MacFaden, and A. Royar. 2014. A Versatile, Production-Oriented Approach to High-Resolution Tree-Canopy Mapping in Urban and Suburban Landscapes Using GEOBIA and Data Fusion. Remote Sensing 6:12837–12865.

O'Neil-Dunne, J. P. M., S. W. MacFaden, A. R. Royar, and K. C. Pelletier. 2013. An object-based system for LiDAR data fusion and feature extraction. Geocarto International 28:227–242.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12:2825–2830.

Tucker, C., M. Brandt, P. Hiernaux, A. Kariryaa, K. Rasmussen, J. Small, C. Igel, F. Reiner, K. Melocik, J. Meyer, S. Sinno, E. Romero, E. Glennie, Y. Fitts, A. Morin, J. Pinzon, D. McClain, P. Morin, C. Porter, S. Loeffler, L. Kergoat, B.-A. Issoufou, P. Savadogo, J.-P. Wigneron, B. Poulter, P. Ciais, R. Kaufmann, R. Myneni, S. Saatchi, and R. Fensholt. 2023. Sub-continental-scale carbon stocks of individual trees in African drylands. Nature 615:80–86.

Van Der Walt, S., J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu. 2014. scikit-image: image processing in Python. PeerJ 2:e453.