PROJEKTARBEIT

des Studiengangs Informationstechnik
der Dualen Hochschule Baden-Württemberg Mannheim

THEMA

Introduction to Automated Classification of Remote Sensing Data using Airborne LiDAR Measurements

---

Timo Salisch

18. Mai 2023

Bearbeitungszeitraum:	18.04.2023 – 18.05.2023
Matrikelnummer, Kurs:	4785113, TINF20IT1
Ausbildungsbetrieb:	Deutsches Zentrum für Luft- und Raumfahrt (DLR)
Abteilung:	Methodik der Fernerkundung – Earth Observation Data Science
Betreuer:	Dr. Conrad M Albrecht
Unterschrift Betreuer	

Erklärung

Ich versichere hiermit, dass ich meine Projektarbeit mit dem Thema

Introduction to Automated Classification of Remote Sensing Data using Airborne LiDAR Measurements

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Oberpfaffenhofen, den 18. Mai 2023

Introduction to Automated Classification of Remote Sensing Data using Airborne LiDAR Measurements

Timo Salisch
DHBW Mannheim
Mannheim, Germany
s200396@student.dhbw-mannheim.de
Student Id: 4785113

***Abstract*—AutoGeoLabel, a method for automatically creating labels for geospatial data, is explained together with an example usage of it. By applying rules to e.g. statistical features of data, AutoGeoLabel can create segmentation maps of unclassified data. It is furthermore demonstrated and evaluated using LiDAR data. Additionally, the benefit of using morphological filters with AutoGeoLabel is shown and a simple approach for finding fitting parameters for these filters is mentioned.**

I. INTRODUCTION

Over the last decades, remote sensing as a whole and especially land use and land cover classification have become more popular (Fig. 1). In general, remote sensing makes use of multiple technologies to observe the environments present on planet Earth. As with many other fields of science, remote sensing is profiting from the increasing amounts of data available [2]. Big data enables the application of deep learning methods for remote sensing which have been shown to outperform traditional machine learning approaches, e.g. for semantic segmentation [3]. A big sub-field of machine learning, supervised learning, requires corresponding labels for the training data. Because labeling the available data is expensive in regard to the (human) effort it requires, ways of automatically generating these labels are researched. One such approach is AutoGeoLabel [4]. In this work, AutoGeoLabel is explained as well as an application of it [5]. Building on this, a demonstration of AutoGeoLabel is presented and evaluated.

A. LiDAR technology

One of the technologies applied in remote sensing is Light Detection and Ranging (LiDAR) [6], [7]. LiDAR can be used with many different platforms, such as aircraft, drones, or satellites. It works by emitting a light (laser) pulse and measuring the reflected pulse. Fig. 2 visually demonstrates an example setup for an airborne LiDAR measurement. Three information that can be measure with LiDAR are:

- Count – Number of returned pulses
- Reflectance – Intensity of a returned pulse
- Elevation – Distance of the object that reflected the pulse

How the count and reflectance information are measured can be seen in Fig. 2 and the elevation value can be computed from the time span between the emission of the pulse and the detection of the returned pulse.

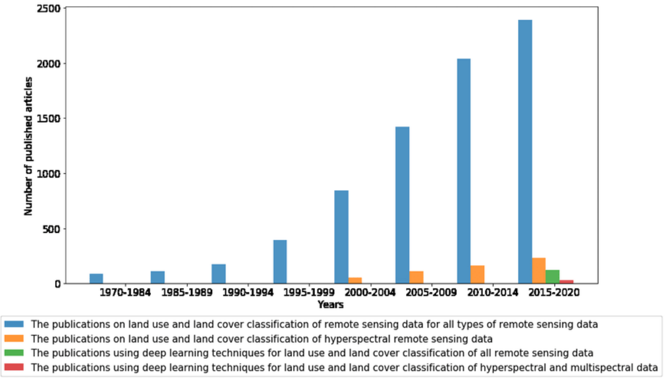


Fig. 1: Amount of publications on land use and land cover classification (source [1])

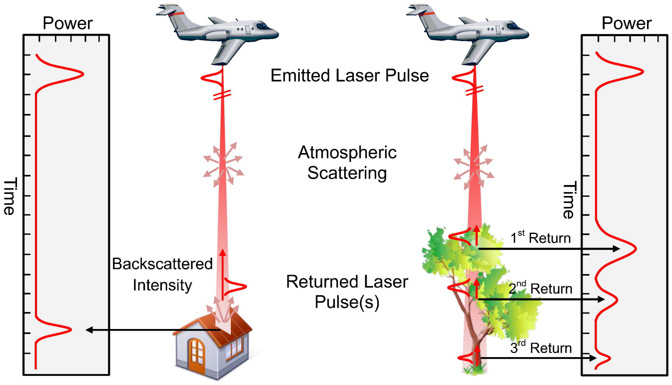


Fig. 2: Airborne LiDAR measurement (source [6])

B. Urban mapping from remote sensing

Urban mapping is one application of remote sensing, where a map of (primarily) urban areas is created using remote sensing data [8] – [11]. It is done by assigning a label or class to every position/pixel of the data. Depending on the data used, different methods or algorithms can be applied to generate the maps. Such urban maps typically contain information about which areas are cover by buildings, roads, vegetation, or other types of objects. These maps can be used for a variety of different applications, see Fig. 3. Depending on the application, different objects might be of interest, resulting in a different set of labels used for the map. The assessment of a city's vegetation might require the classification of different tree species or the distinction between trees and bushes, whereas for self-driving cars or traffic control different road types might be of interest. An example urban map is displayed in Fig. 4.

C. Deep learning/semantic segmentation

Urban maps can be generated using multiple methods, such as pixel-wise classification (applying a method or model to each individual pixel) or semantic segmentation. Semantic segmentation uses a whole image or scene to generate a map for it, which assigns a label to each pixel indicating to what object group the pixel belongs [12] – [15]. An example of an image with a corresponding segmentation map is presented in Fig. 5. An urban map can also be considered to be a segmentation map of an urban area. Deep learning, a subfield of machine learning utilizing deep artificial neural networks, has become a widely used approach for semantic segmentation. This is, because it has been shown to perform better than traditional machine learning methods [3]. Especially convolutional neural network (CNN) based models perform well. They work by compressing the information of the scene into a smaller feature space, and then generating the segmentation map from that smaller set of features. An example can be seen in Fig. 6.

II. WEAKLY-SUPERVISED LEARNING FOR URBAN MAPPING

A. The Concept of AutoGeoLabel

Albrecht et al. (2021) propose a framework for creating labels from raw geospatial data using simple statistical features. To use remote sensing data for machine learning, a more complex (pre-)processing of the data is necessary than with e.g. traditional photography. But in order to use geospatial data for real time applications, labels need to be generated on the fly. To achieve this, the framework AutoGeoLabel is presented, and its effectiveness is shown using LiDAR data. AutoGeoLabel is not limited to LiDAR data, but can be applied to other remote sensing data as well.

The standard process of labeling such data involves human (re-)classification of the data, which causes multiple problems:

- Limited spatial coverage
- Limited temporal updates (once every few years)
- Limited number of classes

On the other hand, OpenStreetMap provides many land classification labels which can be used for training deep neural networks. But there is a large amount of unlabeled geospatial data available, which needs to be labeled applying domain expertise to become applicable for machine learning approaches. AutoGeoLabel might be useful for creating these labels using little computational resources.

For validating the framework, LiDAR data collected in 2017 is used. This data has been classified into eight classes with more than 90% accuracy. These labels are used as ground truth for the validation of the labels created by AutoGeoLabel.

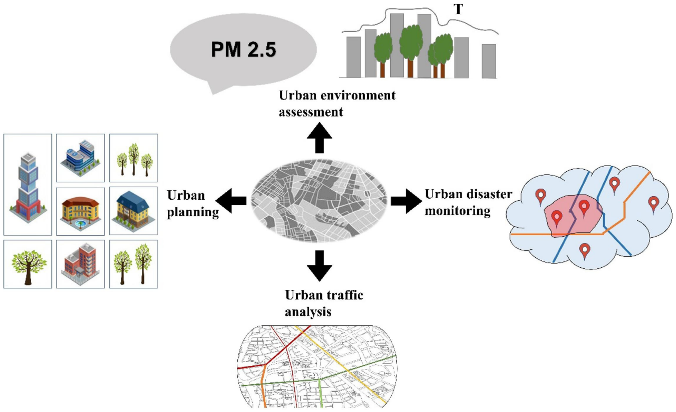


Fig. 3: Possible applications of urban maps (source [8])

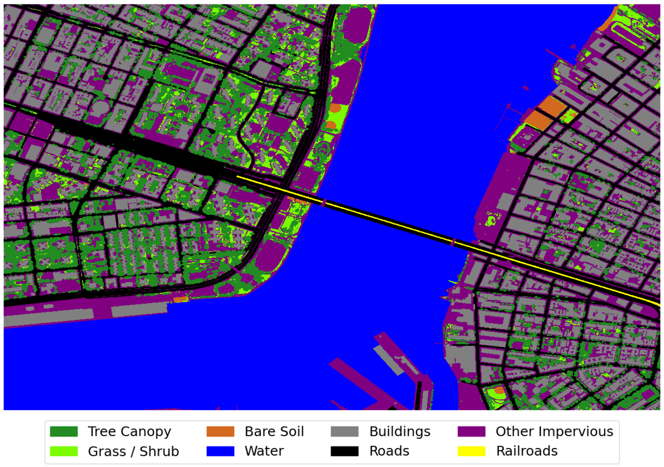
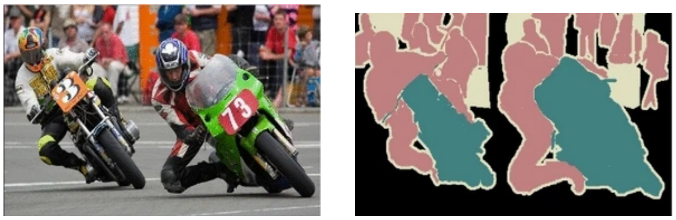


Fig. 4: Urban map of part of New York City

AutoGeoLabel uses statistics of the LiDAR data instead of the raw data. To compute the statistics, the raw data are converted to a nested grid. Using a kernel, the local statistics are generated from the nested grid for the reflectance, count, and elevation information of the LiDAR data. The statistics generated include minimum, maximum, mean, standard deviation, and, for count only, sum. These 13 features are used by AutoGeoLabel for labeling.

Rules are proposed, which use these features, to classify buildings, vegetation, and roads. For the LiDAR data used by the authors, the rules are:

- Buildings: The minimum elevation has to be higher than the average minimum elevation, the standard deviation of the elevation has to be less than the average standard deviation of the elevation, and the maximum elevation is bigger than the average maximum elevation
- Vegetation: The maximum count is bigger than the average maximum count, the standard deviation of the elevation is bigger than the average standard deviation of the elevation, and the standard deviation of the count is bigger than the average standard deviation of the count
- Roads: The minimum reflectance is more than 10% of the maximum reflectance of the complete scene, the mean



(a) Original image (b) Segmentation map

Fig. 5: Example of a semantic segmentation (source [13])

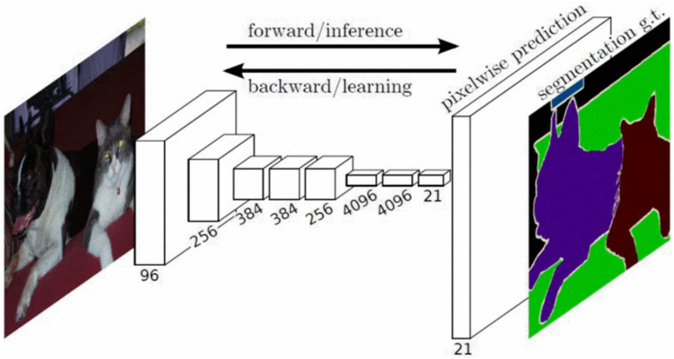


Fig. 6: Example neural network for semantic segmentation (source [12])

reflectance is less than 60% of the maximum reflectance of the whole scene, and the minimum elevation is less than 10% of the maximum elevation of the scene

AutoGeoLabel and the used rules are validated with five different metrics:

- Precision P: $\frac{TP}{TP+FP}$
- Recall R: $\frac{TP}{TP+FN}$
- F_1-score: $\frac{2PR}{P+R}$
- Accuracy acc: $\frac{TP+TN}{TP+FP+TN+FN}$
- Intersection over union IoU: $\frac{|C \cap T|}{|C \cup T|}$

The basis for validation are the values of true positives (TP), false positives (FP), true negatives (TN), false negatives (FN), set of pixels labeled as a class (C), and set of pixels of a class according to the ground truth (T). Table I shows the validation results achieved by the authors. The accuracy score indicates that most pixels are classified correctly, thus a qualitative reconstruction of the scene is possible.

The three classes are well separated in the 13 feature dimensions, as demonstrated by a t-Distributed Stochastic Neighbor Embedding. But because they are not easily linearly separable, simple algorithms, such as K-Means, are not effective. Therefore, more complex methods, such as artificial neural networks with e.g. self-supervised learning, are needed and could be tested in the future.

One example application of AutoGeoLabel is to automatically classify trees to calculate the amount of carbon stored in them using unlabeled LiDAR data. Additionally, it can create labels on the fly in extreme situations, where there either are no labeled data available or the labels are no longer valid. These by AutoGeoLabel created labels can be used to e.g. organize missions in flooded areas.

TABLE I: Validation results from [4]

class	P	R	F_1-score	acc	IoU
buildings	0.98	0.62	0.76	0.88	0.61
vegetation	0.52	0.60	0.55	0.90	0.38
roads	0.91	0.44	0.59	0.93	0.42

TABLE II: Model performance from [5]

Model	Label	precision	recall	F_1	IoU
U-Net	exact	0.85	0.81	0.82	0.71
	noisy	0.87	0.60	0.69	0.55
ref. noisy vs. exact		0.52	0.60	0.55	0.38
SVM	exact	0.82	0.54	0.62	0.48
	noisy	0.77	0.58	0.63	0.48

B. Natural Hazard Monitoring for Hurricane Sandy in New York City

Albrecht et al. (2022) present a method for automatically detecting trees from remote sensing data with little human interaction and evaluate it using data from New York City. Automated label generation, such as AutoGeoLabel, create noisy labels, which may not be precise enough for using them on its own. But they can be used for training machine learning models. This is demonstrated with the example of identifying trees in multi-spectral imaginary using noisy labels created from LiDAR data to determine the damage caused by hurricane Sandy in New York City. For this approach, little human interaction is needed.

LiDAR data from 2017 is used to create the noisy labels. The geospatial area of the LiDAR data has been mapped to eight classes: Tree Canopy, Grass / Shrub, Bare Soil, Water, Buildings, Roads, Other Impervious, and Railroads. This mapping is used as ground truth for the validation of the trained models. The National Agriculture Imaging Program (NAIP) provides orthophotos with four spectral bands (near-infrared, red, green, and blue) from 2011, 2013, 2015, and 2017. The models are trained on the images from 2017 and are then applied to the images from the other years to detect changes over time in the tree canopy.

The (noisy) labels are created from the LiDAR data using AutoGeoLabel. Because the goal is to only classify trees, a mask for trees is created from the noisy labels and used as the labels for training. Using these labels and the NAIP images from 2017, a U-Net and Support Vector Machine (SVM) are trained with data from Queens and then applied to Brooklyn. The U-Net consists of four convolutional down/up sampling layers, and the SVM uses a Gaussian kernel. The by the machine learning models created tree maps for the years 2011, 2013, and 2015 are then compared to evaluate the damages to the trees done by hurricane Sandy in 2012.

Table II presents the results achieved by the models using the noisy labels in comparison to exact labels. The row ”ref. noisy vs exact” indicates the performance assuming the noisy label generation is a model. The results from the models lead to three observations. First, the U-Net performs better than the SVM. Second, both models have a better performance when trained on the exact labels than the noisy labels. And thirdly, the models trained with the noisy labels perform better than the noisy labels itself. Using the U-Net, the impact of hurricane Sandy can be classified into three categories:

- Big loss in trees in suburban areas close to the sea
- Urban areas suffered less loss in trees (the buildings might have protected the trees)
- Suburban areas away from the sea had little to no loss in trees

The images also indicate secondary damages to the trees.
In general, the models achieved comparable results using the noisy labels as with the manually created labels. Future work might investigate how models can be improved to work better with noisy labels. But these models can already be applied to different cities or a different detection of changes to e.g. support other projects.

III. AUTOGEOLABEL DEMONSTRATION

A. Dataset

The LiDAR data used for evaluating AutoGeoLabel has been collected in Williamsburg, New York City (NYC) in 2017 [16]. More than eight pulses per square meter have been used for sampling the data [17]. Instead of the raw data, local statistics derived from the LiDAR data are used. These statistics are created according to [4] for the information elevation and count. Included are minimum, maximum, mean, and standard deviation. To evaluate the achieved results, a ground truth is required. For that, a land cover map that is based on the LiDAR data and developed by the University of Vermont is used. It has a 6-inch resolution and the area of NYC is mapped to eight classes: Tree Canopy, Grass/Shrubs, Bare Soil, Water, Buildings, Roads, Other Impervious, and Railroads. Used for the evaluation are only the classes buildings and tree canopy. Both classes have an accuracy of above 99% in the land cover dataset and are thus accurate enough to be used as ground truth.

B. Implementation

AutoGeoLabel is implemented using the rules from [4] for the classes buildings and vegetation. To improve the results, morphological filters are applied to the predictions of the rules. For the vegetation class, an erosion is used and for the building class opening and closing is applied. A Python implementation for class building is presented in Listing 1 and the implementation for class vegetation in Listing 2. The kernel sizes for the filters are chosen by optimizing the intersection over union (IoU) metric for the data. In order to do so, a grid search is implemented and used. To use the land cover dataset for evaluating the rules with the different configurations of the morphological filters, all classes except vegetation and building are mapped to the class ”other impervious”.

Listing 1: Python implementation of AutoGeoLabel for class building

```
skimage.morphology.binary_closing(
    (
        (featureMap[:, :, 0] > means[0]) &
        (featureMap[:, :, 1] < means[1]) &
        (featureMap[:, :, 2] > means[2])
    ).astype(np.uint),
    np.ones((closing_size, closing_size)
)
```

Listing 2: Python implementation of AutoGeoLabel for class vegetation

```
skimage.morphology.binary_erosion(
    (
        (featureMap[:, :, 0] > means[0]) &
        (featureMap[:, :, 1] > means[1]) &
        (featureMap[:, :, 2] > means[2])
    ).astype(np.uint),
    np.ones((erosion_size, erosion_size))
)
```

C. Evaluation results

The results of the rules only (without any filters applied) are presented in Table III. In comparison to the results from [4] (Table I), the rules achieved a 4% worse IoU score for the building class but a 10% better IoU score for vegetation. For both classes, the accuracy is worse (2% for building and 6% for vegetation). The segmentation map created by AutoGeoLabel for an example scene is displayed in Fig. 7. Comparing to the equivalent ground truth (Fig. 4) reveals that the semantic structure of the scene is contained in the AutoGeoLabel results. But it also shows that some objects are consistently classified into the wrong class, e.g. the railroads on the bridge over the river. To improve the reliability of AutoGeoLabel, morphological filters are applied to the results. For buildings, closing and opening filters are tested and for vegetation erosion. Using a grid search, a filter size of 20 for the closing and no opening for the building class and a filter size of 4 for the erosion of the vegetation class is found to work best. Both filters are squared with a shape of $filter\ size\ x\ filter\ size$. Applying these filters to the results of AutoGeoLabel, the urban map for the scene in Fig. 7 changes to Fig. 8. The evaluation of AutoGeoLabel with the filters is presented in Table IV. This demonstrates, that using the filters, the IoU is increased by 7% for class building and 4% for vegetation. The accuracy is improved for both classes as well (building: 2% and vegetation: 6%). With the filters applied, the results for both classes are better than achieved in [4]. It can thus be concluded, that depending on the data, the use of morphological filters can improve the results achieved by AutoGeoLabel. Fig. 9 visualizes the changes done by the filters and shows that especially the classification of the edges of buildings is improved by both filters, the erosion and the closing. But it also demonstrates that the improvements by

TABLE III: Evaluation results raw rules

class	P	R	F_1-score	acc	IoU
buildings	0.92	0.59	0.72	0.86	0.57
vegetation	0.53	0.85	0.65	0.84	0.48

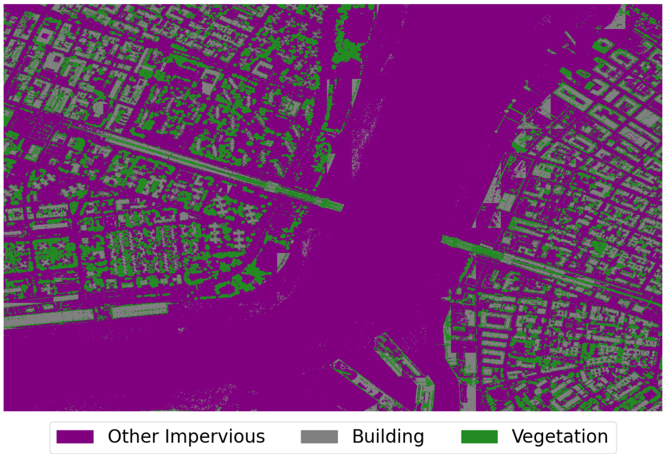


Fig. 7: Urban map created by AutoGeoLabel without filters

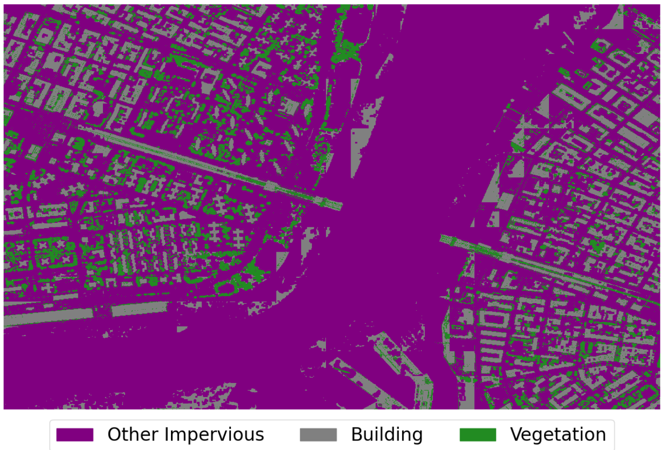


Fig. 8: Urban map created by AutoGeoLabel with filters

TABLE IV: Evaluation results with morphological filters

class	P	R	F_1-score	acc	IoU
buildings	0.85	0.72	0.78	0.88	0.64
vegetation	0.74	0.64	0.69	0.90	0.52

filters are limited. That is because they worsen the results at the same time. Fig. 9a demonstrates this effect, not only the edges of buildings are added but also objects that are falsely classified as building, such as the bridge or sections in the river, are closed by the filter. These filters can therefore only improve the results up to a predefined point and, after which a change in the kernel size starts to decrease the rules' accuracy again. Especially for the LiDAR data, some pixels comply with the rules for multiple classes and other pixels comply only with the rules for classes they are not. The reason for that are similarities in the statistics that lead to wrong or ambiguous classification of pixels. It could be prevented by making the rules more specific, but this then leads to the rules not being generalized enough to capture all pixels of a class. This is because the rules can be seen as hyperplanes in the feature dimension, dividing the pixels into classes. But as argued in [4], the classes are not easily linearly separable and hence such simple (linear) rules cannot achieve perfect results for LiDAR data.

IV. CONCLUSION

Given the amount of unlabeled geospatial data available, AutoGeoLabel can help to automatically create labels for the data. AutoGeoLabel works by manually finding rules for any kind of geospatial data that divide the data into the classes required. These rules could be applied to the raw data or, as demonstrated by the authors of AutoGeoLabel, to statistical features of the data. Depending on the application, the rules can achieve an accuracy above 90%. If the accuracy is not enough, the labels created by AutoGeoLabel can be used to train machine learning models that are able to achieve better results than AutoGeoLabel itself. AutoGeoLabel can therefore be part of a pipeline that is used to train models on unlabeled data without much human effort involved. To improve the accuracy of AutoGeoLabel itself, the usage of morphological filters could be beneficial depending on the data.

REFERENCES

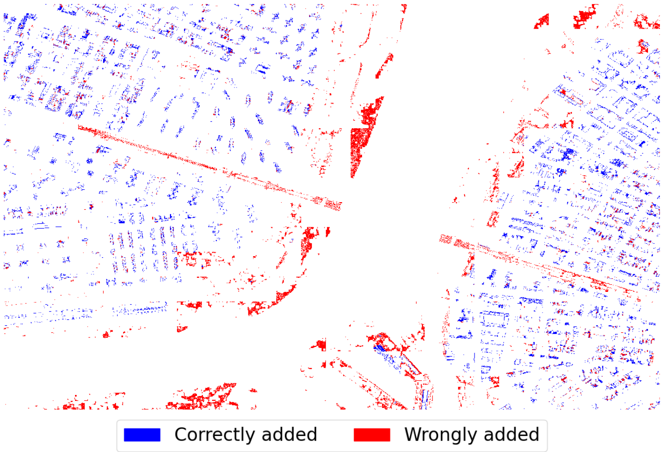
[1] Ava Vali, Sara Comai, and Matteo Matteucci. "Deep Learning for Land Use and Land Cover Classification Based on Hyperspectral and Multispectral Earth Observation Data: A Review". In: *Remote Sensing* 12.15 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12152495. URL: https://www.mdpi.com/2072-4292/12/15/2495.

[2] Peng Liu. "A survey of remote-sensing big data". In: *Frontiers in Environmental Science* 3 (2015). ISSN: 2296-665X. DOI: 10.3389/fenvs.2015.00045. URL: https://www.frontiersin.org/articles/10.3389/fenvs.2015.00045.

[3] Wuttichai Boonpook et al. "Deep Learning Semantic Segmentation for Land Use and Land Cover Types Using Landsat 8 Imagery". In: *ISPRS International Journal of Geo-Information* 12.1 (2023). ISSN: 2220-9964. DOI: 10.3390/ijgi12010014. URL: https://www.mdpi.com/2220-9964/12/1/14.

[4] Conrad M Albrecht, Fernando Marianno, and Levente J Klein. "AutoGeoLabel: Automated Label Generation for Geospatial Machine Learning". In: *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, Dec. 2021. DOI: 10.1109/bigdata52589.2021.9672060. URL: https://doi.org/10.1109%5C%2Fbigdata52589.2021.9672060.

[5] Conrad M Albrecht et al. *Monitoring Urban Forests from Auto-Generated Segmentation Maps*. 2022. arXiv: 2206.06948 `[cs.CV]`.



(a) Changes in class building

(b) Changes in class vegetation

Fig. 9: Changes by applying morphological filters to AutoGeoLabel result

[6] Wai Yeung Yan, Ahmed Shaker, and Nagwa El-Ashmawy. “Urban land cover classification using airborne LiDAR data: A review”. In: *Remote Sensing of Environment* 158 (2015), pp. 295–310. ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2014.11.001. URL: https://www.sciencedirect.com/science/article/pii/S0034425714004374.

[7] Zhien Wang and Massimo Menenti. “Challenges and Opportunities in Lidar Remote Sensing”. In: *Frontiers in Remote Sensing* 2 (2021). ISSN: 2673-6187. DOI: 10.3389/frsen.2021.641723. URL: https://www.frontiersin.org/articles/10.3389/frsen.2021.641723.

[8] Jiadi Yin et al. “Integrating remote sensing and geospatial big data for urban land use mapping: A review”. In: *International Journal of Applied Earth Observation and Geoinformation* 103 (2021), p. 102514. ISSN: 1569-8432. DOI: https://doi.org/10.1016/j.jag.2021.102514. URL: https://www.sciencedirect.com/science/article/pii/S030324342100221X.

[9] Assefa M. Melesse et al. “Remote Sensing Sensors and Applications in Environmental Resources Mapping and Modelling”. In: *Sensors* 7.12 (2007), pp. 3209–3241. ISSN: 1424-8220. DOI: 10.3390/s7123209. URL: https://www.mdpi.com/1424-8220/7/12/3209.

[10] Robbe Neyns and Frank Canters. “Mapping of Urban Vegetation with High-Resolution Remote Sensing: A Review”. In: *Remote Sensing* 14.4 (2022). ISSN: 2072-4292. DOI: 10.3390/rs14041031. URL: https://www.mdpi.com/2072-4292/14/4/1031.

[11] Amir Reza Shahtahmassebi et al. “Remote sensing of urban green spaces: A review”. In: *Urban Forestry & Urban Greening* 57 (2021), p. 126946. ISSN: 1618-8667. DOI: https://doi.org/10.1016/j.ufug.2020.126946. URL: https://www.sciencedirect.com/science/article/pii/S1618866720307639.

[12] Biao Li et al. “A Survey on Semantic Segmentation”. In: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. 2018, pp. 1233–1240. DOI: 10.1109/ICDMW.2018.00176.

[13] Yanming Guo et al. “A review of semantic segmentation using deep neural networks”. In: *International Journal of Multimedia Information Retrieval* 7 (June 2018). DOI: 10.1007/s13735-017-0141-z.

[14] Shijie Hao, Yuan Zhou, and Yanrong Guo. “A Brief Survey on Semantic Segmentation with Deep Learning”. In: *Neurocomputing* 406 (2020), pp. 302–321. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.11.118. URL: https://www.sciencedirect.com/science/article/pii/S0925231220305476.

[15] Fahad Lateef and Yassine Ruichek. “Survey on semantic segmentation using deep learning techniques”. In: *Neurocomputing* 338 (2019), pp. 321–348. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.02.003. URL: https://www.sciencedirect.com/science/article/pii/S092523121930181X.

[16] “Discover GIS Data NY”. In: (). URL: https://orthos.dhses.ny.gov/ (visited on 05/15/2023).

[17] “Understanding the 2017 New York City LiDAR Capture”. In: (). URL: https://maps.nyc.gov/lidar/2017/ (visited on 05/15/2023).