# Open Data and machine learning in the service of complementing municipal GIS inventory

Joel Martin Geda [a], László Zentai [b], Andrea Pődör [c,*]

[a] *University of Óbuda, Alba Regia Technical Faculty, Székesfehérvár, Hungary; joelmartin.geda@gmail.com*

[b] *ELTE Eötvös Loránd University, Institute of Cartography and Geoinformatics, Budapest, Hungary; laszlo.zentai@elte.hu*

[c] *Institute of Geoinformatics, University of Óbuda, Alba Regia Technical Faculty, Székesfehérvár, Hungary; podor.andrea@amk.uni-obuda.hu*

* Corresponding author

**Abstract**: In this study the authors investigated the possibilities to use open data and open software complemented with machine learning to enhance the content of municipal databases. In the study area in Székesfehérvár, a GIS system is used with approximately with 30 modules, although many are still missing. The authors prepared examine the easiest and most affordable methods to extract data to use in two future modules: Parking and Traffic Engineering module. In parking model along field survey, they used QGIS and OpenStreetMap, in the other module they used Google StreetView for defining the places of traffic signs and used machine learning to define the signposts. They found that the accuracy of creating the parking module is based on the completeness of the database and the field measurement method, in case of the Traffic Engineering method the up-to-dateness and completeness of the original data source (Google Street View) and the number of teaching samples influence the results.

**Keywords:** Urban GIS, machine learning, open data

## 1. Introduction

The primary objective of our study was to explore the potential for further development of urban geographic information systems. There are more developed regions where the digital twin of the municipality is already fully established, but in Hungary the average municipality is still relatively far from this level of development. We chose the city of Székesfehérvár as our pilot area, which has a relatively average GIS system.

Nowadays, we are seeing the emergence of open-source software and open data, which are the main resources we have relied on in our research.

There is research, that proved that OpenStreetMap in urban area can be more accurate than authoritative data (Haklay, 2010; Neis et.al., 2010; Helbich et.al, 2012) therefore seemed like a possible solution, and as an open-source database we relied on it in our experiment. There are modules of QGIS which allow us to connect directly to this database (Sehra et al, 2017).

Also, Google Street View (GSV) has long been used for environmental studies (Rundle et al. 2011) because, it has reduced the fieldwork, but there are also many studies that show that it plays an important role in geospatial data collection today (Biljecki et al, 2021). Even traffic signs can be retrieved with the help of GSV (Balali et al, 2015). Recently Campbell, et al, 2019 used GSV combining with deep learning method to automatically extract traffic signs.

### 1.1 Introduction of the recent GIS system of Székesfehérvár

The spatial databases are independent, and their structure is defined by properly designed data models, which can be customized by the user. It is built on a Server-Client architecture, which means that there is a central server that runs it and a client that runs it through an internet browser. It is a modular, versatile, and flexible software architecture (Grósz, 2018).

Its development is ongoing, and is currently in progress, along with change management within the system. Ongoing change management is essential to the usefulness of the system, as it needs to be up to date to be used properly. Change management can be monthly, quarterly, or bi-annual, based on a predefined set of rules.

The application has been built in a modular way, allowing for easier development. It currently consists of more than 30 modules, but some of these have been merged over time. Among the most important modules are the Public Area Tools module, the Redline module, the Search module. The Public Space Tools module, as the name suggests, is dedicated to public space tools.

We have seen that there are several modules that currently support the work of the City's professionals, but there are still more modules that could be added to the existing ones. For example, the Parking, Traffic Engineering and Municipal Property modules.

We explored the possibilities of developing each of these modules using the databases currently available. In the

Proceedings of the International Cartographic Association, 5, 6, 2023.
31st International Cartographic Conference (ICC 2023), 13–18 August 2023, Cape Town, South Africa. This contribution underwent single-blind peer review based on submitted abstracts. https://doi.org/10.5194/ica-proc-5-6-2023 | © Author(s) 2023. CC BY 4.0 License.

current study, we will focus on our findings for the Parking module and the Traffic Engineering module.

## 2. The possible development of the Parking module

For this module, we looked at the easiest and fastest ways to extract data.

In the module, we need to know the size of the parking space, which the software used by the municipality can use to calculate the number of possible spaces in the parking lot. It is important to note here that handicapped spaces are larger than a conventional parking space, so we need to take this into account. The parking lot we have studied as a sample area is in Székesfehérvár, in the Sarló street. The parking lot is 140 meters long and 18 meters wide, approximately 2520 square meters. There are 88 parking spaces in total. As can be seen in Figure 1, this is a small car park, but it is perfect for illustration purposes. The time taken to survey the car park was approximately half an hour in the field.

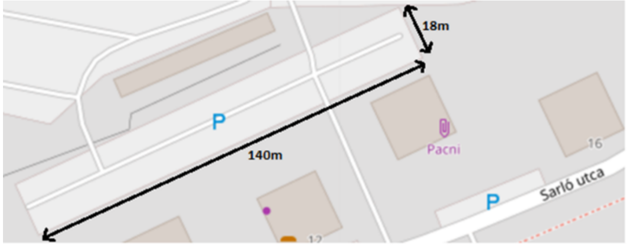


Figure 1. The sample car park.

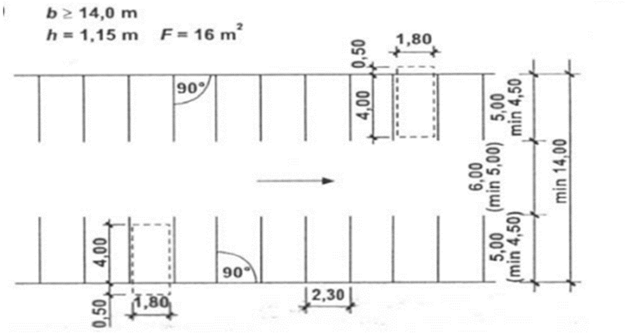


Figure 2. Scaling of parking spaces in the city.

Based on Figure 2, it can be determined that the size of a parking space is 11.5 square metres (5mx2.3m), while the size of the roadway in a 140-metre-long parking space is 1120 square metres (140mx8m), we had to consider that the parking space is crossed by another road of 144 square metres (18mx8m). Also, from the parking lot there is another parking lot and the road that goes with it, which occupies 40 square meters (5mx8m) of the total area, which is 2520 square meters. The road is 8m wide, as the total width of the car park is 18m, but the length of the two car parks must be at least 10m. Thus, a total area of 1216 square meters is occupied by the roads in the parking area, while the remaining 1685 square meters of space can be used for parking. Based on this calculation, the car park should ideally accommodate 105 parking spaces. However, after a field survey, it was found that there were only 88 spaces in the area. The difference between the two values is 16%.

### 2.1 Data extraction based on OSM and QGIS

The parking was also surveyed in QGIS software, using the QuickMapServices and QuickOSM modules. QuickMapServices can be used to find datasets and base maps, while QuickOSM can be used to retrieve data from OpenStreetMap (OSM) from a database in the cloud. The values associated with the key "amenity" were retrieved from the database, providing data for all the car parks in the study area (Fig.3). The Field Calculator now makes it easy to query the areas associated with the parking spaces.

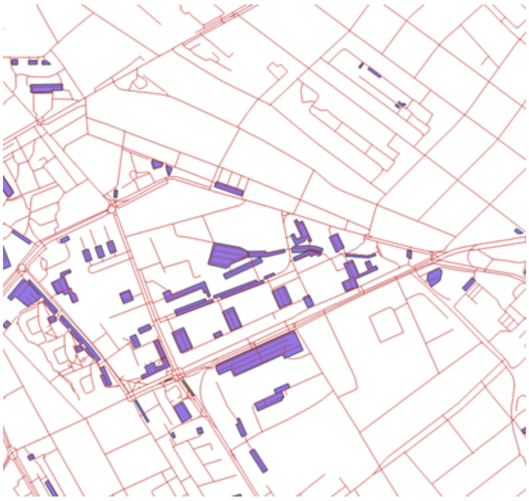
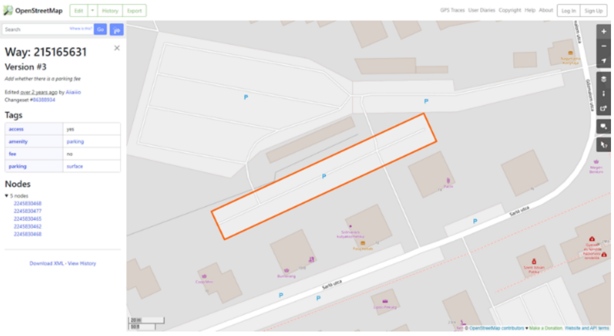


Figure 3. Car parks stored in OSM in the study area.

Unfortunately, the database did not list the parking lot we were looking for by street name, only other parking lots on Sarló Street. Thus, a search by street name in the database did not yield any results. As the database is incomplete, it was necessary to use a manual search in OSM to select the parking lot. Thus, it was possible to identify it and its ID. The database contains an attribute called "osm_id" which contains the unique identifier of the elements. Filtering on this unique identifier (Fig.4.), it was now easy to retrieve all the data related to the car park, including its area.



Way: 215165631

Version #3

Add whether there is a parking fee

Edited over 2 years ago by Aiaiiio
Changeset #86388934

Tags

| | |
|---|---|
| access | yes |
| amenity | parking |
| fee | no |
| parking | surface |

Nodes

▾ 5 nodes
2245830468
2245830477
2245830465
2245830462
2245830468

Download XML · View History

Figure 4. The specific parking place in OSM.

The examined area of the car park under consideration is 2591,428 square metres. Based on the field visit, the area under investigation is 2520 square meters, while the area in the database is 2591,428 square meters. This is a discrepancy of roughly 3% due to the fact that the field survey was not carried out with professional equipment and that we have to take into account that we do not know with which equipment the data stored in OSM were surveyed. The time taken to survey the car park area was approximately one and a half hours, as the database had to be properly prepared for the query. This would take considerably less time for the additional car parks, while the field survey took 30 minutes. This showed that the area counted on the computer could be obtained much more quickly after an initial preparation, but this is only possible if you already have existing data to count with, in this case the OSM database. As the database does not always store parking as descriptive data, a preliminary survey is essential to know the correct data.

## 3. The possible development of the Traffic Engineering/Traffic Sign Detection module

The second module development is the traffic engineering module, which collects the traffic signs located in Székesfehérvár. This module would store in accurate form the information about the number and type of signs in the city. In addition, it is important to know the coordinates of the signs. A sample area was selected to test the development of the module. The area to be tested is the area bounded by Budai út, Gáz utca, Király sor. Basically, two possible solutions were investigated in the study. However, three potential solutions are possible, the first one being the most trivial, namely, to survey the signs on foot and to cluster them in a database. This is an extremely time and resource consuming solution with relevance only in a small study area. This solution is poorly scalable and therefore not investigated in our current study.

### 3.1 Using Google Street View

In the first method investigated, Google Street View was used to mark the signs by placing pins in the Android application called Map Marker. These can be exported to several file formats from the app. For example, in .kmz, .csv and .xlsx formats, making it easy to import the database into other software. Based on Google Street View, on 2.5 km long road there are approximately 193 signs, and it took a total of 105 minutes to survey (Fig.5.). It should be noted that Google Street View does not provide an up-to-date picture in most cases. For this reason, the survey is to be carried out in combination with the city-walk as described below. With the first solution considered, the company would receive a pre-surveyed database containing a significant part of the signs. This would mean that only the missing signs would have to be added to the database or those no longer there would have to be removed.

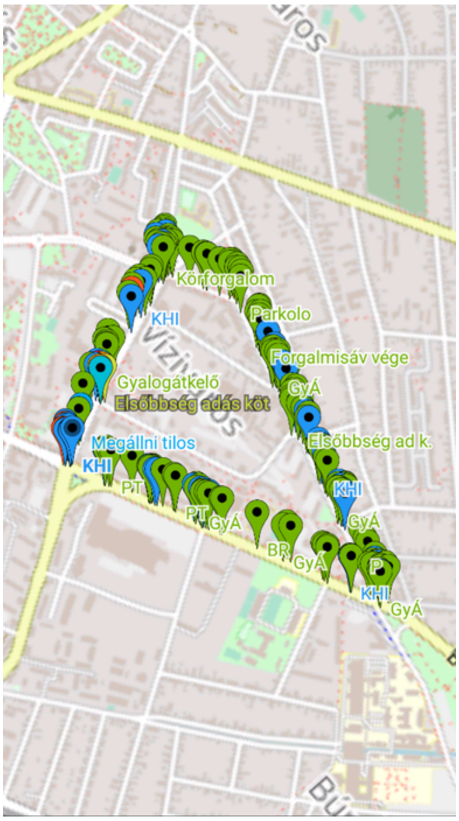


Figure 5. Map Marker for traffic sign marking.

### 3.2 Machine learning in detecting traffic signs with GSV

The second solution was implemented using machine learning and a slightly modified C# code from Simon Parkinson. (Dakin et al, 2020) Here, the C# code essentially launches the browser and Google Streetview. It then retrieves the coordinates of the positions to be examined from a predefined database. Then it takes 360-degree screenshots of the received positions and saves them. Taking these photos lasted about half an hour. We then created a supervised machine learning model using the Google Cloud and Vertex AI. To train the model, 100 images were used from the images extracted using the method. In case of a supervised model, we must manually label a sample dataset to specify our target output data. (Fig.6.).

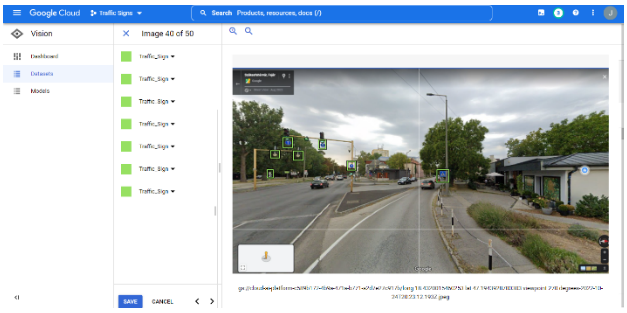
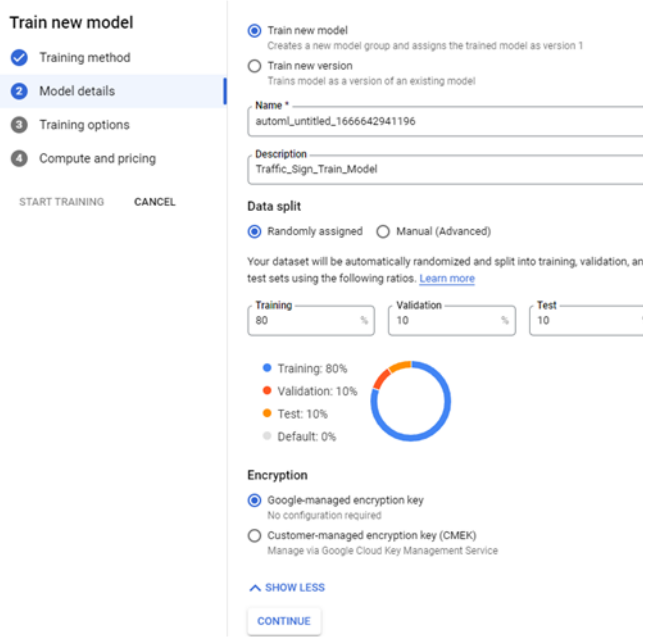


Figure 6. Labelling the traffic signs.

After determining our scope for the model, the software separates the sample dataset into three groups, training, validation, and test groups (Fig.7.). For a hundred images this means that there will be eighty in the training group, ten in the validation group and ten in the test group. The training time took roughly one hour. Of course, the more images used for training, the more accurate the model is (Gad, 2018).


Train new model

1. Training method
2. Model details
3. Training options
4. Compute and pricing

● Train new model
Creates a new model group and assigns the trained model as version 1
○ Train new version
Trains model as a version of an existing model

Name *: automl_untitled_1666642941196

Description: Traffic_Sign_Train_Model

Data split

● Randomly assigned ○ Manual (Advanced)

Your dataset will be automatically randomized and split into training, validation, an test sets using the following ratios. Learn more

| Training | Validation | Test |
|---|---|---|
| 80 % | 10 % | 10 |

- Training: 80%
- Validation: 10%
- Test: 10%
- Default: 0%

Encryption

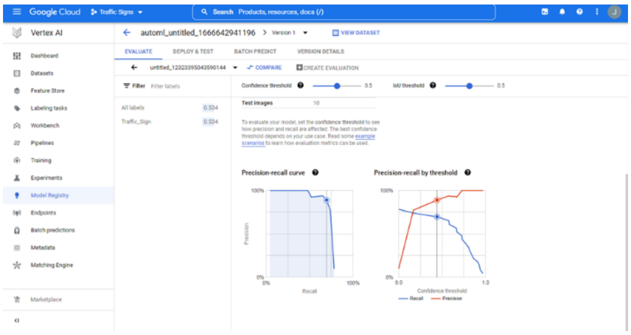
● Google-managed encryption key
No configuration required
○ Customer-managed encryption key (CMEK)
Manage via Google Cloud Key Management Service

Figure 7. The three groups of the test images.

Google's basic payment model is called "pay as you go", so the test was done using a minimum number of images.

## 4. The results of the machine learning model

Once the training is complete, Google provides summary statistics, with a graph of the Precision-recall curve and a graph of the Precision-recall by threshold. The Precision-recall graph provides an immediate, visual picture of the performance of the selected category. If the line is at the top right of the graph, the selected category is performing well. If the line is at the bottom left of the graph, it indicates that the category is performing poorly. In the current case, the graph shows that the category performed well during the training session. The Precision-recall by threshold graph (Fig. 8.) displays the recall and accuracy rates of the selected category based on the test results.



Confidence threshold 0.5
IoU threshold 0.5

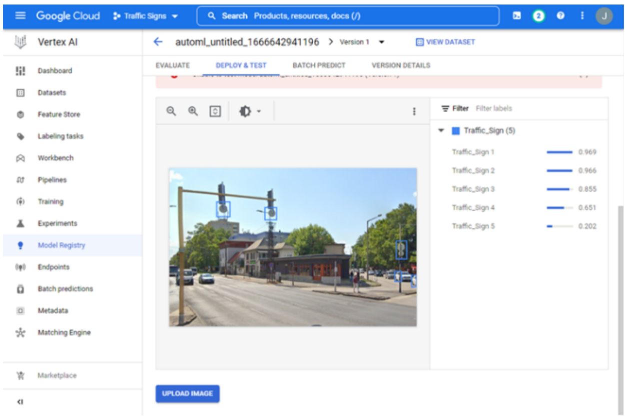
| | |
|---|---|
| All labels | 0.534 |
| Traffic_Sign | 0.534 |

Test images 10

To evaluate your model, set the confidence threshold to see how precision and recall are affected. The best confidence threshold depends on your use case. Read some example scenarios to learn how evaluation metrics can be used.

Figure 8. The graphs of the training model.

The graph can be used to determine the thresholds for a category or to examine the performance of the selected category. The blue line shows the recall rate for different threshold settings. The red line shows the accuracy rate for different threshold settings. For a given category, the graph shows the relationship between accuracy and recall at different thresholds. The higher the threshold, the higher the accuracy, but the lower the recall. The ideal threshold setting is the highest possible recall and accuracy rate. This goal is not always achieved because the higher the recall rate, the lower the accuracy rate, and vice versa. The most appropriate threshold setting for a category is a compromise between these two ratios. If the accuracy rate is high under strict scoring requirements, the content classification will miss many items that belong to that category. For accuracy, the system will not include many items that should be included in the category. However, if the recall rate is high and many items are received, accuracy will be lower.



Traffic_Sign (5)

| | |
|---|---|
| Traffic_Sign 1 | 0.969 |
| Traffic_Sign 2 | 0.966 |
| Traffic_Sign 3 | 0.855 |
| Traffic_Sign 4 | 0.651 |
| Traffic_Sign 5 | 0.202 |

Figure 9. Testing the model.

What can be seen from such a small sample is that the model is able to recognise traffic signs, but because the sample is too small, in some cases it also marks other objects, but for these it says with less than 80% probability that it is the traffic sign (Fig.9.).

## 5. Conclusion

In our study we investigated the possibility for enriching municipal GIS inventory based on Open data with open software. We explored OpenStreetMap and Google Street View as a potential data source for municipal Geographical Information System. As our results shows OSM and GSV may provide a good starting point for urban use, but field verification is essential for checking timeliness and accuracy. However, a detailed field survey can be replaced by the methods presented here or by deep learning methods previously developed specifically for traffic sign collection (Balali et al, 2015 and Campbell et al, 2019). The methods we have tested have proven to be effective, but as our examples have shown, there may be discrepancies in terms of accuracy and completeness of the descriptive data in the database (OSM), and for both OSM and GSV, it is necessary to check the date of updating the database and pictures. However, for the creation of the initial data of the database of the planned modules, they are fully adequate, but for accuracy and geometric verification, we see at this stage that human manual intervention is necessary.

31st International Cartographic Conference (ICC 2023), 13–18 August 2023, Cape Town, South Africa. This contribution underwent single-blind peer review based on submitted abstracts. https://doi.org/10.5194/ica-proc-5-6-2023 |

## 6. References

Balali, V., Ashouri Rad, A., & Golparvar-Fard, M. (2015). Detection, classification, and mapping of US traffic signs using google street view images for roadway inventory management. Visualization in Engineering, 3(1), 1-18.

Biljecki, F., & Ito, K. (2021). Street view imagery in urban analytics and GIS: A review. Landscape and Urban Planning, 215, 104217.

Campbell, A., Both, A., & Sun, Q. C. (2019). Detecting and mapping traffic signs from Google Street View images using deep learning and GIS. Computers, Environment and Urban Systems, 77, 101350.

Dakin, K., Xie, W., Parkinson, S. et al. Built environment attributes and crime: an automated machine learning approach. Crime Science 9, 12 (2020). https://doi.org/10.1186/s40163-020-00122-9

Gad, A. F., (2018). Practical computer vision applications using deep learning with CNNs. Berkeley: Apress.

Grósz Gábor (2018) Városgazda és INVATER Térinformatikai Rendszer Felhasználói kézikönyv Budapest, HungaroCAD Informatikai Kft.

Haklay, M. (2010), How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. Environment and Planning B, 37, pp. 682- 703. (19) [accessed Jan 04 2023].

Helbich, M., Amelunxen, C., Neis, P., & Zipf, A. (2012). Comparative spatial analysis of positional accuracy of OpenStreetMap and proprietary geodata. Proceedings of GI_Forum, 4, 24.

Neis, P.; Zielstra, D.; Zipf, A. and Struck, A. (2010) Empirische Untersuchungenzur Datenqualität von OpenStreetMap - Erfahrungen aus zwei Jahren Betriebmehrerer OSM Online-Dienste. in Strobl, J. et al. (Eds.): Angewandte Geoinformatik 2010: Beiträge 22. AGIT-Symposium Salzburg, 2010, pp. 420-425.

Rundle, A. G., Bader, M. D., Richards, C. A., Neckerman, K. M., & Teitler, J. O. (2011). Using Google Street View to audit neighborhood environments. American Journal of Preventive Medicine, 40(1), 94-100.

Sehra, S. S., Singh, J., & Rai, H. S. (2017). Assessing OpenStreetMap data using intrinsic quality indicators: an extension to the QGIS processing toolbox. Future Internet, 9(2), 15.

31st International Cartographic Conference (ICC 2023), 13–18 August 2023, Cape Town, South Africa. This contribution underwent single-blind peer review based on submitted abstracts. https://doi.org/10.5194/ica-proc-5-6-2023 | © Author(s) 2023. CC BY 4.0 License.