

A WEBGIS FRAMEWORK FOR SEMI-AUTOMATED GEODATABASE UPDATING ASSISTED BY DEEP LEARNING

R. Can, S. Kocaman, A.O. Ok

Dept. of Geomatics Engineering, Hacettepe University, 06800 Beytepe Ankara, Turkey –
(recepcan, sultankocaman, ozgunok)@hacettepe.edu.tr

Commission V, WG V/3

KEY WORDS: WebGIS, Citizen Science, Deep Learning, Geodatabase Updating, Change Detection

ABSTRACT:

The automation of geoinformation (GI) collection and interpretation has been a fundamental goal for many researchers. The developments in various sensors, platforms, and algorithms have been contributing to the achievement of this goal. In addition, the contributions of citizen science (CitSci) and volunteered geographical information (VGI) concepts have become evident and extensive for the geodata collection and interpretation in the era where information has the utmost importance to solve societal and environmental problems. The web- and mobile-based Geographical Information Systems (GIS) have facilitated the broad and frequent use of GI by people from any background, thanks to the accessibility and the simplicity of the platforms. On the other hand, the increased use of GI also yielded a great increment in the demand for GI in different application areas. Thus, new algorithms and platforms allowing human intervention are immensely required for semi-automatic GI extraction to increase the accuracy. By integrating the novel artificial intelligence (AI) methods including deep learning (DL) algorithms on WebGIS interfaces, this task can be achieved. Thus, volunteers with limited knowledge on GIS software can be supported to perform accurate processing and to make guided decisions. In this study, a web-based geospatial AI (GeoAI) platform was developed for map updating by using the image processing results obtained from a DL algorithm to assist volunteers. The platform includes vector drawing and editing capabilities and employs a spatial database management system to store the final maps. The system is flexible and can utilise various DL methods in the image segmentation.

1. INTRODUCTION

The spatial and semantic updating of geodatabases containing land use land cover (LULC) information is a crucial process to ensure their usability. The spatial data updating can be considered as a two-step process: (i) accurate detection of the area of change, and (ii) precise determination of the modified geometry due to the change. These steps have traditionally been performed by mapping professionals with the help of aerial/satellite/UAV imagery or via fieldwork. Nowadays, many volunteers and citizen scientists contribute to the geodata collection and interpretation activities, such as OpenStreetMap (OSM) Project with over seven million contributors (OSM, 2021). In this respect, rapid and accurate image data collection is crucial for emergency response planning and mitigation in numerous circumstances including the emergencies caused by disasters occur due to seismic activities, floods, landslides, wildfire, etc. However, as the size of the area mapped by images increases, the interpretation and analysis of images become labour intensive, costly and time consuming. Therefore, provision of timely data and their interpretation is still a vivid research area; and in many cases, such data can be rapidly collected with a certain level of quality with the help of citizen scientists and volunteers, who often have little or no knowledge on image processing or map-updating.

Deep learning (DL) architectures, especially deep convolutional neural networks (CNNs), have increasingly been used for semantic segmentation/classification of airborne imagery (e.g. Wu et al., 2018; Bittner et al., 2018; Yang et al., 2018; Shi and Zhu, 2018). Many state-of-the-art DL architectures (Ronneberg et al., 2015; He et al., 2017; Chen et al., 2018) have shown

outstanding performances in the segmentation/classification tasks, if sufficiently enough training datasets are supplied to the DL architecture. However, their outputs are still not been frequently utilised or preferred for updating a geodatabase in an end-to-end framework. Nonetheless, such DL-based methods have great potential to assist both experts and citizen scientists in a semi-automated manner by recommending tags for classification, verifying tags and updates, supporting quality control check for output data, detecting and monitoring pixel-based changes (immediately) after natural hazards etc. Facilitating the DL techniques for updating a geodatabase would noticeably empower such interpretation tasks mostly done by manual processing, particularly for revealing areas with change. In this way, the time and personnel costs required can be reduced significantly, and such interactive approaches will ensure more accurate and semantically correct data in a relatively shorter amount of time.

Although DL-based approaches can significantly enrich the intelligence while updating a geodatabase, the DL architectures can also benefit from the volunteered geographical information (VGI) by the training data collected with the help of mobile and web-based Geographical Information Systems (WebGIS) platforms (Chen and Zipf, 2017). Fan et al. (2021) proposed an interactive platform for 3D building modelling from VGI data. Integrated with the increased processing power of the mobile devices, the DL techniques and WebGIS have great potential for accurately and timely managing, analysing and presenting geospatial data. Thanks to the expansion in computer technology and its requirements, the WebGIS platform architecture is evolving recurrently. In this context, several review studies summarise the work carried out till now (e.g.

Agrawal and Gupta, 2017; Rowland et al., 2020) and interested readers may refer to those studies for further details. Such platforms can also effectively provide DL-based decision support systems to assist decision makers through interactive tools. In a study conducted by Can et al. (2019) showed that the data quality issues in citizen science (CitSci) data collection projects can be mitigated with the help of CNNs. In a recent work, Can et al. (2020) presented a WebGIS framework for the integration of the developed CNN-based quality assessment method in a GeoAI platform (called GeoCitSci.com).

In this study, a geospatial artificial intelligence (GeoAI) supported by a WebGIS platform was designed and implemented to demonstrate how DL can aid geodatabase updating especially for LULC data. The platform developed in this study currently supports for one feature type, i.e. polygons utilised for building roofs/footprints. However, thanks to its modular design, the system can be easily expanded to other types of objects, and the other use-case scenarios by training and executing further DL models. Therefore, our main contribution in this study is to proposing a flexible general purpose WebGIS platform to be utilised for GeoAI purposes.

The structure of this paper is organized as follows. The system design and implementation are described in the following section. The results and the related discussion are presented in Section 3. Finally, in Section 4, we give concluding remarks and make suggestions for possible future works.

2. PROPOSED SYSTEM DESIGN

The developed GeoAI platform principally requires a vector file containing building boundaries to be employed as ground-truth and a single airborne image to be used in the DL-based image segmentation. Thereafter, the platform processes the input data using DL architecture. The output of our platform is the changes detected as the vector file, which is finally presented to the user of the platform through a WebGIS interface. The user can easily navigate to the locations with detected changes listed through the interface. By clicking on a specific vector element labelled as difference (or change), the map is automatically zoomed to the location of the selected difference area. The user can also observe the old and new data on a specific location. The WebGIS platform enables the user to add, modify and delete features through a map editor. Any modifications submitted by the users are immediately transferred to the geodatabase for updating. The users have possibility to download, and locally store the vector map data any time until the processing session ends.

The operational system workflow and the main component of the system and their interactions between the different technologies are presented in Figures 1 and 2, respectively. As shown in Figure 2, the system includes four major components, i.e. *web map interface*, *change detection*, *geospatial analysis*, and *data management*. The component *web map interface* includes functionalities that enable the user to observe and update areas with changes detected via the DL assistance. The component *change detection* includes an automated pipeline with the help of a DL module requiring an input of a single airborne image with related georeferencing information; and producing an output georeferenced vector file with the detected buildings. The component *geospatial analysis* enables spatial analysis functionalities. The component *data management* is responsible for data input-output processes, the data transactions, and the management of the spatial database

management system (SDBMS) and the file system. The system components and technologies presented in Figure 2 are briefly described in the following sections.

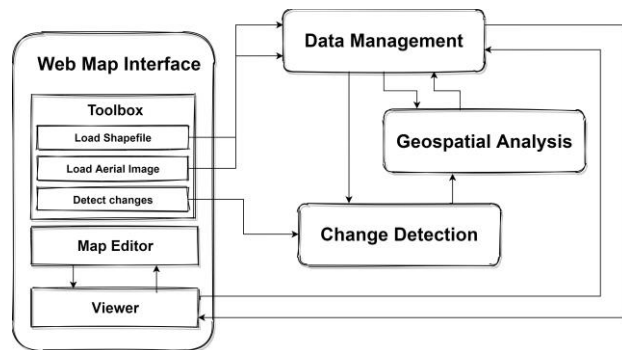


Figure 1. The system workflow.

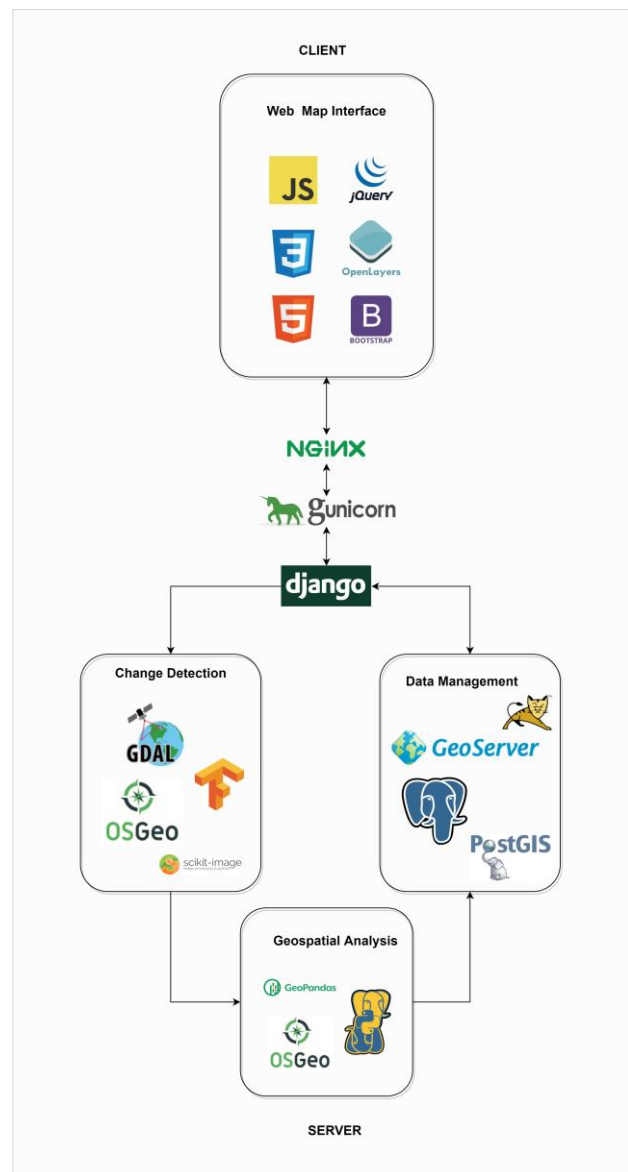


Figure 2. The system components and technologies utilised.

2.1 Web Map Interface

Keeping the easiness and simplicity are of our major offer when designing the web map interface, so that inexperienced users would be able to use the system without the necessity of any prior knowledge or expertise. The interface is divided into three sections, which are the toolbox, map editor and viewer. The map editor shows the functionalities that the user can operate, specifically functions to add, edit and delete data. The map editor menu can only be accessible if a vector file (i.e. shapefile) is uploaded in active session. The viewer section is responsible for showing active layers, which are base map layers, an airborne image and a vector data file that are provided by the user, and the resulting detected changes layer obtained from the DL module. At this time, only OSM (OSM, 2021) and Bing Maps Aerial (Microsoft, 2021) are provided to serve as base maps. Besides, the viewer section enables the user to perform selected functionality in the map editor menu. The toolbox enables the user to choose the operations to be performed, such as loading the shapefile and airborne image or processing a change detection procedure, etc. OpenLayers (OpenLayers, 2021), jQuery and JavaScript are used for the viewer section, and an open-source front-end library, Bootstrap (Bootstrap, 2021), is used for the development of the interface.

2.2 Change Detection

The change detection component of the system includes an automated pipeline. First, this component pre-processes the airborne image uploaded by the user. The aim of the pre-processing is preparing and forming the input airborne image for the DL module. Once the pre-processing is completed, the component creates a configuration file to be used by the DL module, and calls it. Our DL module currently includes a modified version of RA-Unet (Jin et al., 2020), which was pre-trained using a subset of Inria Image Labeling Dataset (Maggiori et al., 2017) within this study. The DL module locates the configuration file with the pre-processed airborne image, and finally produces a georeferenced raster comprising the segmentation result produced by the modified RA-UNet. Within the component, raster to vector conversion approach proposed by Sahu and Ohri (2019) was employed to produce a vector file from the segmentation results. The modified version of RA-UNet was implemented by using TensorFlow (Abadi et al., 2015). In our framework design, GDAL (GDAL/OGR contributors, 2021), Fiona (Gillies et al., 2011), Shapely (Gillies et al., 2007), PyProj, scikit-image (Walt et al., 2014), OpenCV (Bradski, 2000) and SciPy (Virtanen et al., 2020) libraries were used for raster and vector data manipulations.

2.3 Geospatial Analysis Component

The geospatial analysis component is responsible for discovering the differences between the ground truth vector data provided by the user and the segmentation results, which were produced by the change detection component. The geospatial analysis component initially searches any overlap between the geometric entities in two datasets. Next, it extracts the differences (which are in principle polygons) that exist in the segmentation results but not in the ground truth vector data. After this step, a size threshold calculated using the ground truth geometries is applied to the detected differences. If the size of the difference area is smaller than the threshold, the area is removed from the list of detected changes. The final product of the geospatial analysis component is the output georeferenced vector data showing the final differences between

the ground truth vector data and the segmentation results. Geopandas, GDAL and PyProj libraries were used for the implementation of geospatial analysis component and vector data processing functionality.

2.4 Data Management Component

The data management component is responsible for data input-output processes, data transactions, and the management of the SDBMS and the file system. When the user uploads a zip file through "Load Shapefile" form in the web map interface, the related function of the component is triggered. This function applies several operational evaluations before writing the shapefile to the database. If the shapefile is successfully qualified from all available predefined tests such as geometry validation, coordinate reference system control, etc., the shapefile is written to the database and immediately published on Geoserver map engine from the Open Source Geospatial Foundation (Geoserver, 2021). Also, the user can upload an airborne image through "Load Aerial Image" form under the toolbox on the web map interface. Similar to the shapefile function of the component, the aerial image function checks the uploaded aerial image and publishes the image on Geoserver. The change detection and the geospatial analysis components also exploit the data management component for writing the data to the database or the file system. The component utilises PostgreSQL with PostGIS extension as SDBMS for storing vector data, and Geoserver for sharing both the vector and raster data types.

3. RESULTS AND DISCUSSION

The main idea behind this study is to demonstrate how DL can assist the tasks for change detection based map updating. Therefore, we designed and implemented the system as simple as possible to focus on the overall framework. Our system can be modified for different tasks and can be adapted to different conditions. The DL model remains in the core of the system, and various DL models can also be utilised depending on the problem defined and the datasets available. The DL models, especially for airborne image segmentation tasks, require well prepared datasets and demands for high computational resources in order to train and fine tune the models to increasing the prediction performances. If the computational resources are limited, the model training from the scratch may last several days to months without anticipating the success of the final model or the configuration. The model training is an iterative process; one needs to fine tune hyper-parameters, and to modify the model until the desired performance is achieved. In this study, we carried out a segmentation model developed in the medical image processing domain, and adjusted the model in order to utilise with airborne images. Since our computational resources are not quite high, only a base model was trained for this study by using a subset of the Inria Image labelling dataset (ca. 20% with 36 images for training and 10 images for testing).

Examples from the DL output and the interface elements of the implemented GeoAI platform are presented in Figures 3-13. In Figure 3, the menu for loading vector and image data is shown. In Figure 4, the layer selection menu and the vector data fetched from the OSM are depicted. Some of the buildings existed in the OSM data were purposefully removed for testing reasons. Figure 5 shows the map editor menu. In Figures 6 and 7, the web map editing and application of the changes in the database are demonstrated, respectively. Figure 8 shows an aerial image from Inria Image Labelling dataset, which was not used in

training or validation during the training of the demo model. In Figures 9 and 10, the menu for starting the DL model for change detection and the outputs as a list of areas (polygons) with detected changes is demonstrated, respectively. The results can also be displayed with a layer on/off functionality (Figures 10 and 11). Figure 11 also shows one example of successfully detected missing building in the vector data provided by the user. The building can then either be drawn from scratch, or the polygon, which is the output of the DL segmentation, can be modified by the user. In Figure 13, the overall DL segmentation results, and the download functionality for the updated vector data are depicted.

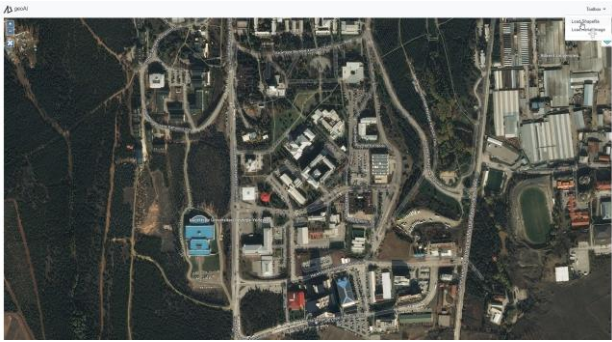


Figure 3. WebGIS interface elements of the developed GeoAI platform.

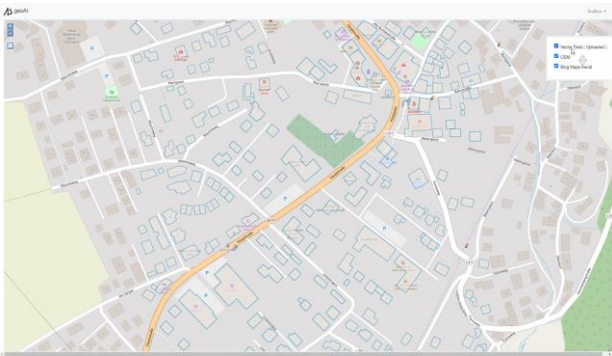


Figure 4. Layer switcher to control the layer visibility by the user.



Figure 5. The map editor menu, which can be opened by clicking the tools symbol below the zoom icon.



Figure 6. Vector editing functionality on the WebGIS interface.



Figure 7. All modifications applied on the uploaded vector data are committed in the spatial database immediately after clicking the finalize symbol.



Figure 8. The layer switcher menu updated after loading the test aerial image from the Inria Image labelling dataset.



Figure 9. Detect changes menu, which triggers the change detection component by applying the DL method.

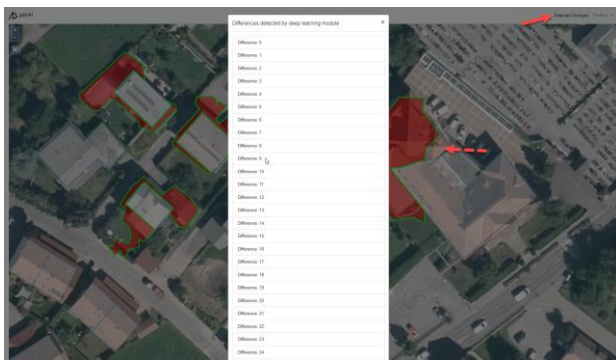


Figure 10. The list of detected change polygons returned by the DL method which also allows navigating to the locations.

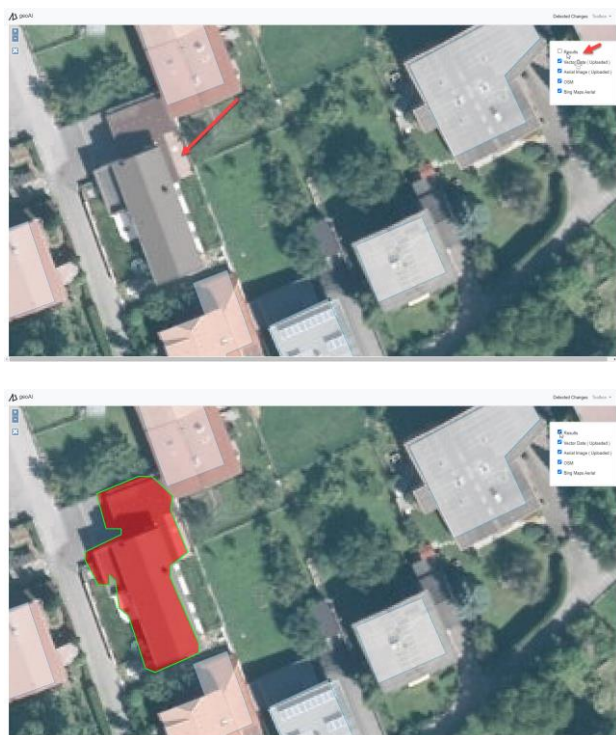


Figure 11. The layer switcher menu updated after the change detection is completed, and the location of one missing building in the ground truth (above), and the DL segmentation result below.

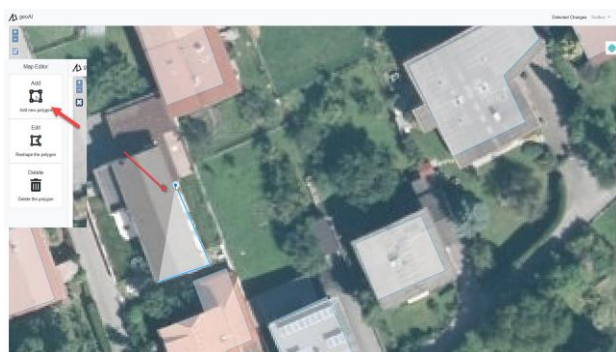


Figure 12. Map drawing tool on the WebGIS interface for the missing building.

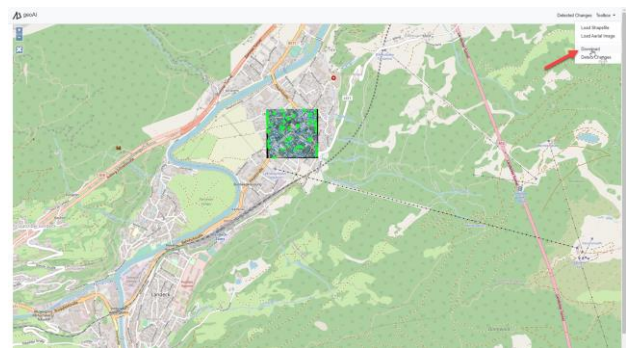


Figure 13. An overview of DL segmentation result on the test image and vector download menu for the updated vector data.

Since we only trained a base DL model for a few epochs and a small subset of the Inria Image Labelling dataset, the model performance is not quite high, and the changes detected currently suffer from erroneous segmentation output. Thus, the improvement of the DL model for increasing the prediction performance remains an open task. Besides, as stated before, other DL models can be integrated into the developed system for problem specific tasks. The input raster image resolution must also be taken into account when feeding into the change detection component. Other parameters like the image acquisition angles (i.e. nadir and off-nadir), atmospheric conditions, the date and hour of image acquisition, the geographic location of images would also play a role on the success of building change detection. Furthermore, large georeferencing errors in aerial images would also distort the results and must be taken into consideration as an additional processing step.

Considering the completeness and the scalability of the developed system; it can be deployed by mapping agencies or geospatial companies with further system optimizations and after handling the security related issues. However, if the system is deployed at global scale, data input-output standards must also be considered since they may cause considerable amount of difficulties. Furthermore, the performance related issues such as the number of concurrent users to run the change detection component or to modify their vector data, and the number of requests that the servers can handle must also be considered by the related agencies/companies.

4. CONCLUSIONS AND FUTURE WORK

In this study, we designed and implemented a geospatial artificial intelligence (GeoAI) supported WebGIS platform to demonstrate how DL can aid geodatabase updating. The proposed DL-assisted WebGIS framework has numerous application areas, especially when the data is urgently needed or the data quality, in particular the completeness, is of high importance. Platforms such as OSM or mapping agencies, geospatial companies can also adapt the proposed methodology in their image processing frameworks to improve the operations. Extra features, such as periodical scans and warning on the images for detecting the areas that require updating can also be included as add-ons in the developed application.

The developed GeoAI system is considered to be integrated to the GeoCitSci.com, a CitSci platform for geoscience researches developed at Hacettepe University as joint efforts of Geomatics and Geological Engineering Departments (Kocaman and Gokceoglu, 2019; Can et al., 2019; Can et al., 2020). Thus,

further DL algorithms for the characterization of various geomorphological characteristics and geohazards can be also utilised in the system.

ACKNOWLEDGEMENTS

This research is conducted as a part of the M.Sc. thesis of the first author.

REFERENCES

Abadi, M., Agarwal, A., Barham, P. et al., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. www.tensorflow.org (25 April 2021)

Agrawal, S., Gupta, R.D., 2017. Web GIS and its architecture: a review. *Arabian Journal of Geosciences*, 10(23), 1-13. <https://doi.org/10.1007/s12517-017-3296-2>

Bradski, G., 2000. The OpenCV Library. Dr. Dobb's Journal of Software Tools.

Bootstrap, 2021. <https://getbootstrap.com/> (25 April 2021).

Bittner, K., Adam, F., Cui, S., Körner, M., Reinartz, P., 2018. Building footprint extraction from VHR remote sensing images combined with normalized DSMs using fused fully convolutional networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(8), 2615-2629. <https://doi.org/10.1109/JSTARS.2018.2849363>

Can, R., Kocaman, S., Gokceoglu, C., 2019. A Convolutional Neural Network Architecture for Auto-Detection of Landslide Photographs to Assess Citizen Science and Volunteered Geographic Information Data Quality. *ISPRS International Journal of Geo-Information*, 8(7), 300. <https://doi.org/10.3390/ijgi8070300>

Can, R., Kocaman, S., Gokceoglu, C., 2020. Development of a CitSci and artificial intelligence supported gis platform for landslide data collection. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, 43, 43-50. <https://doi.org/10.5194/isprs-archives-XLIII-B5-2020-43-2020>

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L., 2018. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 40 (4), 834–848. <https://doi.org/10.1109/TPAMI.2017.2699184>

Chen, J., Zipf, A., 2017. DeepVGI: Deep learning with volunteered geographic information. In Proceedings of the 26th International Conference on World Wide Web Companion, April, pp. 771-772. <http://dx.doi.org/10.1145/3041021.3054250>

Fan, H., Kong, G., & Zhang, C. (2021). An Interactive platform for low-cost 3D building modeling from VGI data using convolutional neural network. *Big Earth Data*, 5(1), 49-65. <https://doi.org/10.1080/20964471.2021.1886391>

GDAL/OGR contributors, 2021. GDAL/OGR Geospatial Data Abstraction Software Library. Open Source Geospatial Foundation. <https://gdal.org> (25 April 2021)

Geoserver, 2021. <http://geoserver.org/> (25 April 2021)

Gillies, S. et al., 2007. Shapely: manipulation and analysis of geometric objects. toblerity.org. <https://github.com/Toblerity/Shapely> (25 April 2021)

Gillies, S. et al., 2011. Fiona is OGR's neat, nimble, non-nonsense API. [Toblerity](http://toblerity.org). <https://github.com/Toblerity/Fiona> (25 April 2021)

Kocaman, S. Gokceoglu, C., 2019. A CitSci app for landslide data collection. *Landslides*, 16, 611. <https://doi.org/10.1007/s10346-018-1101-2>

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, 2961-2969.

Jin, Q., Meng, Z., Sun, C., Cui, H., Su, R., 2020. RA-UNet: A hybrid deep attention-aware network to extract liver and tumor in CT scans. *Frontiers in Bioengineering and Biotechnology*, 8, 1471. Doi: 10.3389/fbioe.2020.605132

Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P., 2017. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In IEEE International Geoscience and Remote Sensing Symposium (IGARSS), July, pp. 3226-3229. <https://doi.org/10.1109/IGARSS.2017.8127684>

Microsoft, 2021. Bing Maps APIs. <https://www.microsoft.com/en-us/maps/choose-your-bing-maps-api> (last accessed on 24.04.2021)

OpenLayers, 2021. <https://openlayers.org/> (25 April 2021).

OSM, 2021. OpenStreetMap Project Contributors. <https://wiki.openstreetmap.org/wiki/Contributors> (31 January 2021).

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: convolutional networks for biomedical image segmentation. In Medical image computing and computer-assisted intervention (MICCAI), volume 9351 of LNCS. New York, NY: Springer, 234–241. Available on arXiv:1505.04597 [cs.CV].

Rowland, A., Folmer, E., Beek, W., 2020. Towards Self-Service GIS—Combining the Best of the Semantic Web and Web GIS. *ISPRS International Journal of Geo-Information*, 9(12), 753. <https://doi.org/10.3390/ijgi9120753>

Sahu, M. Ohri, A., 2019. Vector Map Generation from Aerial Imagery Using Deep Learning. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, IV-2/W5, 157-162. <https://doi.org/10.5194/isprs-annals-IV-2-W5-157-2019>, 2019

Shi, Y., Li, Q., Zhu, X. X., 2018. Building footprint generation using improved generative adversarial networks. *IEEE Geoscience and Remote Sensing Letters*, 16(4), 603-607. <https://doi.org/10.1109/LGRS.2018.2878486>

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M. et al., 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17(3), 261-272. <https://doi.org/10.1038/s41592-019-0686-2>

Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Guillard, E., Yu, T., scikit-image

contributors, 2014. scikit-image: Image processing in Python. <https://doi.org/10.7717/peerj.453>

Wu, G., Shao, X., Guo, Z., Chen, Q., Yuan, W., Shi, X., Xu, Y., 2018. Automatic Building Segmentation of Aerial Imagery Using Multi-Constraint Fully Convolutional Networks. *Remote Sensing*, 10(3), 407.

Yang, H., Wu, P., Yao, X., Wu, Y., Wang, B., Xu, Y., 2018. Building extraction in very high resolution imagery by dense-attention networks. *Remote Sensing*, 10(11), 1768.