Cross verification of revenue using remote sensing data

Technical report

Murtaza Taj
Muhammad Fareed Zaffar

Cross Verification of Revenue Using Remote Sensing Data

Technical Report

Contents

1. Data Objective:

There is presently limited use of Geographic Information System (GIS) and Management Information System (MIS) data available from third parties to help the Government in growing its tax revenue base. This leads to inefficient or arbitrary revenue targets and financial planning, erroneous data leading to inaccurate analysis and various other inefficiencies. It also leads to lack of transparency with regards to spending and resource utilisation, leaving openings for corruption in revenue collection. This project aimed to reduce the unreliability of the current data by using public third-party data providers. Integration of Third-Party GIS and MIS data within the Governments' own GIS and MIS data to identify inconsistencies such as missing properties was proposed. This will help in generating additional revenue for the government and create transparency with regard to budget preparation, spending, and resource utilisation. The pilot phase is applicable for selected regions in Khyber Pakhtunkhwa (KP) and later can be expanded to other parts of the country.

2. Approach

This project involved three stages. In the first stage, the team of Technology for People Initiative (TPI) conducted a change analysis via satellite imagery of the last 3 years starting from 2021, to highlight regions for the physical survey. The KP province conducts a property tax survey every 3 years, but the process is a very slow and labor intensive one. Through our analysis we were able to tell the government which areas of KP have had the most changes in land use (i.e., from vacant land to residential) so that they can focus efforts of their physical surveyors in areas with most land use changes, thus making the use of physical surveyors more efficient and increasing property tax revenue. The findings of this first component were fed into the second component, where the plan was to combine the MIS data with public GIS data available via third-party services and build tools to process and analyse the data. This includes a tool for identification of individual plots within a residential block and a tool for extraction of various plot related parameters such as build-up or vacant, coverage area and height. In future, this stage will help update existing government records and ensure that property taxation is conducted based on the current state of land use. Finally, TPI created a software system for the KP Excise, Taxation and Narcotics Control (ET&NC) Department to track taxation cases. This software integrates remote sensing-based analysis and creates cases based on plot-level inconsistencies between GIS and MIS data. This software will increase data accuracy as well as transparency and accountability for all participants of the system.

3. Methodology

In order to extract meaningful information from GIS data and to allow the ET&NC to make informed decisions, TPI worked on three main image processing modules namely i) Landuse and Land Classification ii) Plot Boundary Extraction and iii) Interactive tool for building height extraction. Each of these modules is discussed next.

3.1. Landuse and Land Classification

Landuse classification of satellite imagery into known classes such as road, houses, vegetation, etc. particularly using high resolution imagery is another area that has greatly benefited from the power of modern machine learning techniques particularly Convolutional Neural Networks (CNNs). Among these architectures, CNN with residual connections (ResNets) based approach has been shown to produce favourable results for the problem of Landuse classification. In this task we used our previous work in which we develop a 14 class
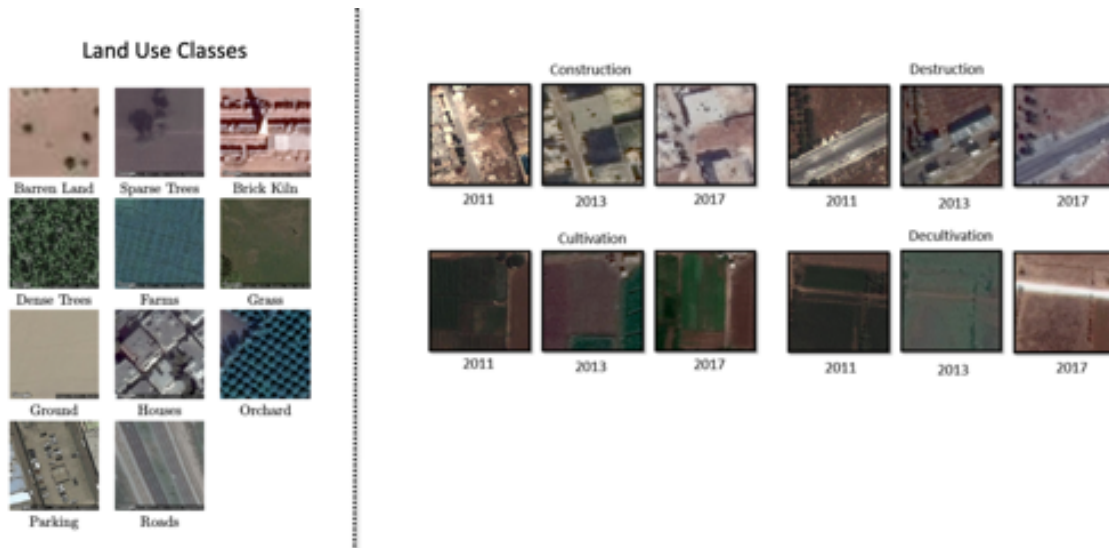
Figure 1: Asia14 dataset containing 14 spatial and 4 temporal classes

dataset namely Asia14 dataset which contains classes such as houses, roads, parking, barren land etc. In our previous work we also developed a solution for the classification of satellite images into one of these 14 classes as well as classification of multi-year satellite imagery into one of the four transition classes namely construction, destruction, cultivation and harvesting. The first classifier is used to extract regions that may be missing in the taxation database but are identified as constructed ares in the satellite imagery. The second classifier is used to update records in the taxation database for the plots where the land use has changed in the past years thus enabling correct tax calculation.

3.2. Plot Boundary Extraction

Information about building block boundaries is integral to the working of the framework. It provides an estimate of the block shape and as a result, helps in hierarchical segmentation within the block. To gather this information, we used a two-fold process, in which we imported manually annotated polygons using KML format from Google Earth Pro and then use QGIS software to extract satellite imagery for each block based on the geo-coordinates from the KML files. Using these polygons as essential data, we designed a machine learning-based solution that automatically extracts plot boundaries. The major steps of the solution are discussed next.

3.2.1.   Step 1: Semi-automatic data annotations

The first step in data-driven machine learning is to collect an ample amount of data so that machines can learn the required task. Thus, in this first step, the existing real estate maps were used to generate plot boundary annotations. First, the geo-referencing of the maps is performed by manually assigning GPS coordinates to at least 3 points on the map, this resulted in the automatic assignment of GPS Coordinates to all the plot boundaries present in the map. Additional annotations were also performed by adding a polygon layer via Google Earth Pro.
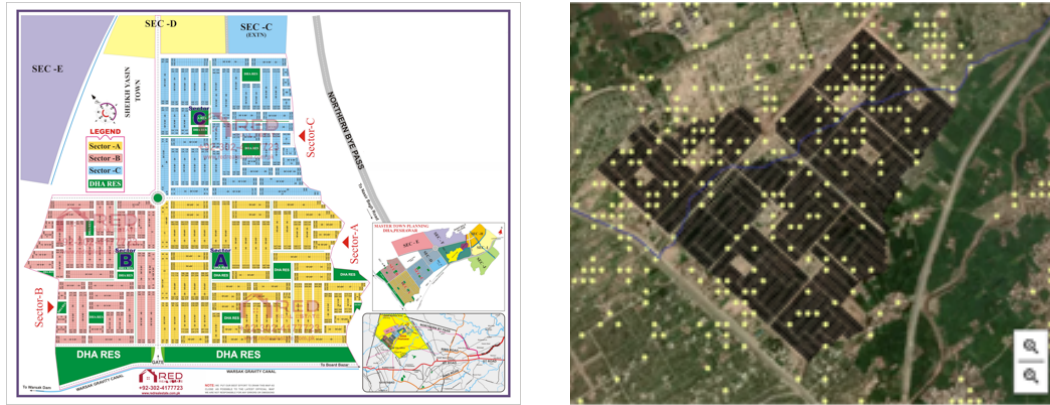
Figure 2: Real estate map and extracted geo-referenced plot boundaries

Figure 3: Google Earth Pro based annotations

3.2.2. Step 2: Instance Segmentation of Plots.

Using the annotations we obtained in the previous step, we trained a pixel-by-pixel labelling algorithm named Mask R-CNN. This machine learning-based approach first localise plots in the scene and then labels the area of plot covered by the houses. As can be seen in the below mentioned figure that due to lack of training data and changing appearance of housing structures within a city, Mask R-CNN has only been successful in finding limited plots within a housing block. This problem is addressed by applying grid fitting as explained in the next step.

Figure 4: Plot extraction using Mask R-CNN

3.2.3. Step 3: Plot boundary refinement

In any housing block, each plot is placed next to another one and each plot has been provided with access to the road. Considering these assumptions we deployed a grid estimation algorithm. This is the final step of the plot boundary extraction module, where detection results from Mask-RCNN are used to estimate an accurate segmentation of buildings within a provided block image. For accurate estimation of the grid structure we use combine basic statistical heuristics and a loss minimization function, that help estimate the structural components of the building block. Before we begin the calculations, we assume that the building structures detected inside the block are of the same length and width. This basic assumption provides us with the liberty to ignore any false negatives i.e. the buildings that Mask-RCNN was unable to detect within a block or the detections that were excluded since they occurred on the edges of a window. As a result, only a few accurate detections within the block are enough to estimate the structure of the entire block.

This technique takes as input two parameters, the length of the building block which is actually the length of the major-axis of the block polygon calculated in the first module. The second parameter are the array of masks detected by Mask-RCNN for the block, converted to contour objects. In the first step, the initial number of vertical divisions inside the block is calculated by dividing the total length of the block by the mean width of all detected masks. This naive estimate is then refined via error minimisation.
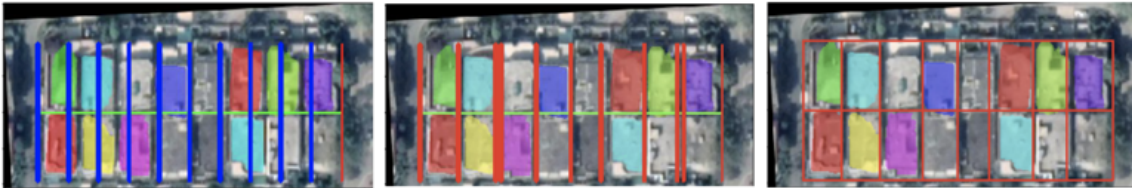

Figure 5: (a) **Initial divisions calculated by dividing total length of the block with mean width of all detected masks. (b) Grid divisions replotted after error minimisation. (c) Complete grid after adjustments**

3.2.4.   Step 4: Housing block extraction and the overall process.

To completely automate the process of plot boundary extraction we utilise google maps. Through basic image processing on Google Maps we first extract housing blocks. These housing blocks are then overlaid on the satellite imagery to extract imagery corresponding to each housing block. The extracted image is then passed to the Mask R-CNN for segmentation of houses followed by grid estimation to extract plot boundary for each plot in the housing block.
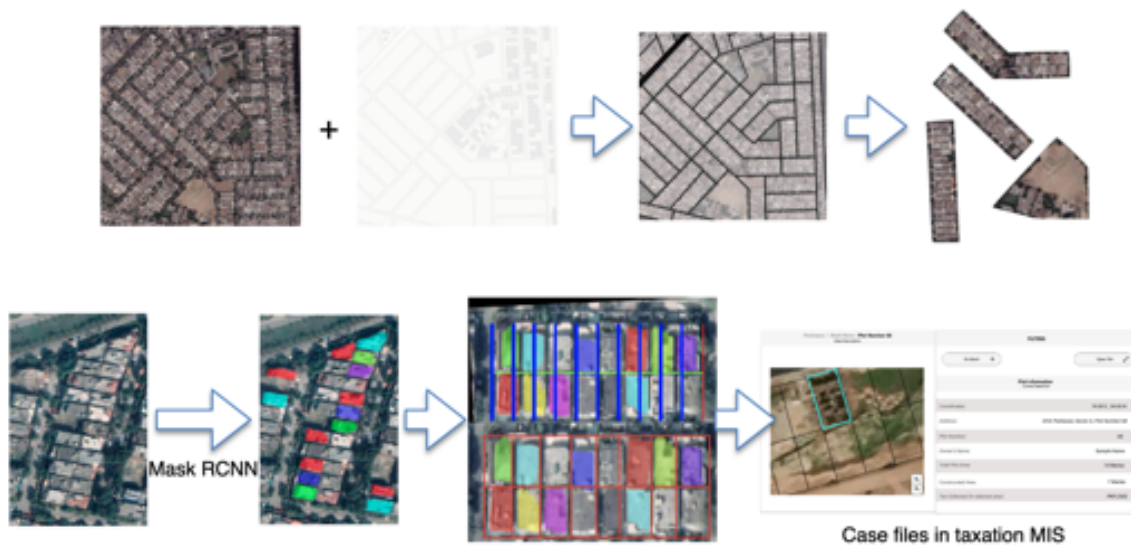

Figure 6: Fully automated process for plot boundary extraction

3.3.   Interactive tool for building height extraction

Building height estimation is approach is inspired from the is broken down into a three-step process, wherein, step one is to test the suitability of the image obtained from the google earth for further processing, step two is to obtain all necessary variables, and step three is to obtain a ratio between shadow length and actual height of buildings for the selected satellite image.

3.3.1.   Step 1: Image Suitability Test

Test for building and shadow: at least one vertical edge of the building can be clearly seen, and shadow for this building must be clear and unambiguous.
Test for slope angle: positive slope should be less than 3 degrees and negative slope should be less than 2.5 degrees for estimation error of less than 5%.
Test for solar azimuth: solar elevation and the solar azimuth of every spot in the same RS image should be roughly the same (proven true for the Google Earth Images).

3.3.2. Step 2: Obtain Necessary Variables

The following variables are obtained directly from the Google Earth application:
- Latitude
- Acquisition date
- Length of shadow
- Length of the vertical edge of the building
- Satellite Azimuth
- Solar Azimuth

The remaining variables were calculated using the equations and corresponding code, where RHS is the ratio of height of a building and length of the vertical edge of the building, and RCS is the ratio of the height of building and length of the shadow cast by building.

3.3.3. Step 3: Height Estimation

With RHS and RCS calculated, the height of any building in the remote sensing image can be simply calculated.

3.3.4. Results

The demo allowed a user to load an image taken from google earth. User would manually select vertical edges of a building and the program would display estimated height next to collection, with all calculations being done in real time. This tool will help the operator in measuring valuable building attributes while assessing the case file for a particular plot. An image of the campus of the Lahore University of Management Sciences was taken to run the demo. Results are shown in the following image:

Figure 7: Image showing height estimated by the proposed solution for various building within the LUMS campus.

4. Software

Based on the above methodology, a software system is created for the KP ET&NC Department to track taxation cases. This software actively takes up machine learning outputs. It is also useful in navigating occupied and vacant lands as well as works well for height estimation. For the KP ET&NC Department, this software will allow users to open and close cases, as well as assign them to physical surveyors and monitor case progress. Overall, this software will increase data accuracy, transparency and accountability for all participants of the system.

The landing page of the web app consists of two main sections.   A map on the left for an interactive graphical representation and a stats section on the right for text-based detailed information as shown below:
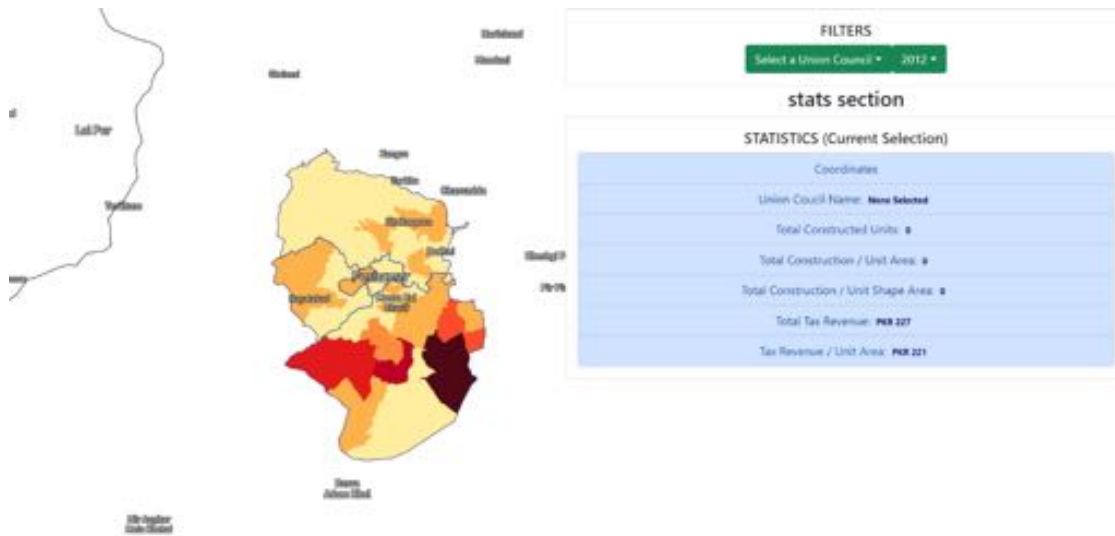

Figure 8: Landing Page of the Web

The map provides an overview of the city of Peshawar. The city can be visually seen divided into union councils. The union councils are coloured according to the activations (constructions) found by our ML model where darker colour indicates more constructions.

The stats-section allows the user to scroll through a dropdown menu as shown in the figure below, that contains the list of all union councils in Peshawar and the years for which data is available.

Figure 9: Dropdown Menu for Union Councils

Selecting one of the union councils and a corresponding year from the menu will highlight that union council on the map and display its corresponding data in the stats section. Initially, the oldest year for which data is available is set as default.

Figure 10: Details of Selected Union Council on the Map

Similarly, selecting a union council from the map will render corresponding information in the stats section.

Figure 11: Stats for the Union Council

Another interface for this webapp and perhaps the most important one builds on the functionality of the aforementioned flow 1 and allows spatial and temporal comparison between union councils. It displays two maps side by side along with their stats sections. So, the user may select different union councils and compare the constructed units between them.
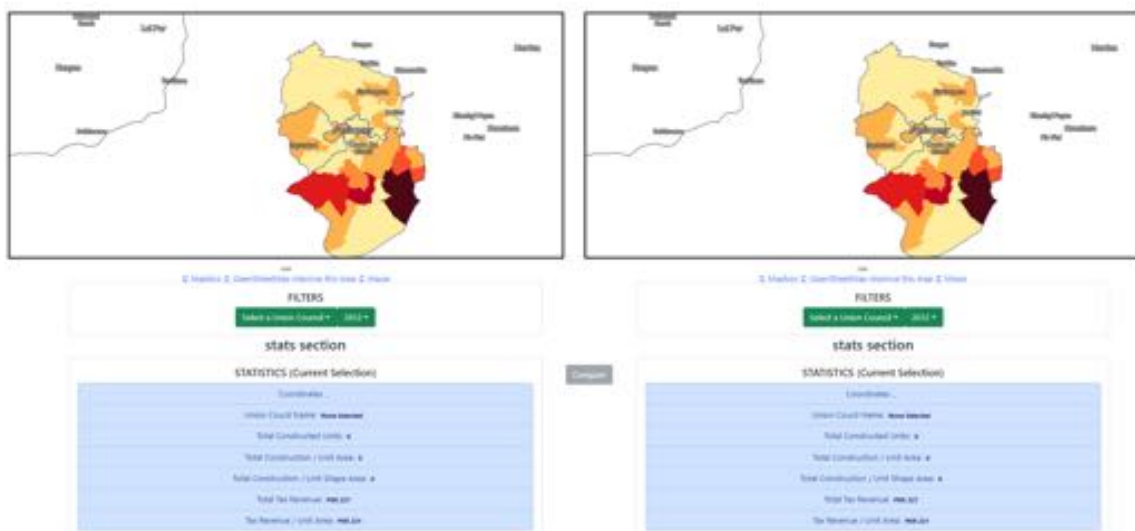

Figure 12: Stats for Comparison

The interface contains a compare button specialised for temporal analysis. It remains disabled until two conditions are met:

1) The same union council is selected for both maps.
2) Different years are selected for both maps.

Upon pressing the compare button after fulfilling the conditions mentioned above a list would open showing the geographical coordinates of the identified constructions that exist in the latter year but not in the previous year. With each identified new construction, four options are available with it, which are outlined hereunder:

1) View File: This will open a file containing satellite imagery as a proof for identified new construction in both years.
2) Accept Proof: The user will confirm the identified new construction by clicking this button and decide upon further action.
3) Reject Proof: The user may reject the identified new construction based on the provided proof by clicking on this button. The system will store this information.
4) Measure Height: This will lead to an interactive solution for real-time building height estimation through satellite imagery.

Figure 13: Four Options for New Construction

5.   Conclusion

Henceforth, aimed at reducing the unreliability of the current data by using public third-party data providers, TPI has been able to bring forth a software that will enhance data transparency to aid governmental operations

The project will result in development of new machine learning technology that will automatically generate land use/landcover maps and will help in identifying regions and locations where land use has changed (for e.g., from no building to building). Identifying these changes will not only help in updating the tax calculations but will also help in registering new locations which are currently not present in the existing MIS. The project will help in identifying the total amount of tax that was not collected in the past, alongside locations that require visit of a surveyor will be automatically identified thus improving the efficiency of the department through effective resource utilisation. The next goal is to test the solution on the Excise and Taxation data and to evaluate and improve the efficacy of the proposed solution through usability test.

6.   References

1. H. Hao, S. Baireddy, E.Bartusiak, M. Gupta, Kevin LaTourette, Latisha Konz, M. Chan, M. L. Comer, E. J. Delp, "Building height estimation via satellite metadata and shadow instance detection", Automatic TargetRecognition XXXI, SPIE 11729, 2021.

2. G. Cheng, C.Yang, X.Yao, L. Guo, and J.Han, "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs," IEEE Trans. Geosci. Remote Sens., vol. 56, no.5, pp. 2811–2821, May 2018.

3. I. Demir et al., "DeepGlobe 2018: A challenge to parse the earth through satellite images," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, Jun. 2018, pp. 172–17209.

4. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

5. M. A. Bhimra, U. Nazir, and M. Taj, "Using 3-D residual network for spatio-temporal analysis of remote sensing data," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2019, pp. 1403–1407.
6. mask-rcnn.

7. Mohanty, S. P. CrowdAI Mapping Challenge 2018: Baseline with Mask RCNN., 2018

8. Nazir, Usman, et al. "Kiln-Net: A Gated Neural Network for Detection of Brick Kilns in South Asia." IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, 2020, pp. 3251–3262.,

9. Q. Liu, R. Hang, H. Song, and Z. Li, "Learning multiscale deep features for high-resolution satellite image scene classification," IEEE Trans. Geosci. Remote Sens., vol. 56, no. 1, pp. 117–126, Jan. 2017.

10. R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," IEEE Trans. Geosci. Remote Sens., vol. 57, no. 8, pp. 5384–5394, Aug. 2019.
11. using mask R-CNN with building boundary regularization". In Proceedings of the IEEE