Land Classification Plugin for QGIS Using Pix2Pix

James Earl Cubillas[1], Reynard Torculas[2], Jessie Mayol[2], Paul Justine Bajao[2]

[1] Virtual Learning Research Center, Caraga State University, Butuan City, Philippines – jdcubillas@carsu.edu.ph
[2] Department of Computer Science, College of Computing and Information Sciences, Caraga State University, Butuan City, Philippines – (reyntors2, mayoljessie2, pauljustine2000)@gmail.com

Keywords: Land Classification, Convolutional Neural Networks, GIS, Unet, QGIS Plugin.

Abstract

Land cover classification is critical in various fields, including environmental monitoring, urban planning, and ecological assessment, facilitating informed decision-making processes. Traditional land cover classification methods often involve labor-intensive and time-consuming processes, relying on manual intervention and predefined algorithms. The emergence of deep learning techniques, particularly convolutional neural networks (CNNs), offers a promising solution to automate this process, albeit with complexities in implementation. This study addresses the limitations of existing Geographic Information System (GIS) software and plugins by proposing a novel approach utilizing the Pix2Pix architecture, a type of CNN, for automated land cover classification. The proposed Land Classification Plugin (LCP) integrates seamlessly with QGIS, offering an end-to-end solution for generating classified static maps. The methodology involves preprocessing data, utilizing the Pix2Pix model for image segmentation, and post-processing to produce georeferenced outputs. The development of the LCP involved extensive software and hardware configurations, including essential components like GDAL/OGR, PyTorch, and OpenCV. The plugin's architecture comprises a user-friendly interface for region selection, clipping, and classification aided by the Pix2Pix model. A layout manager feature also allows for the creation of composite maps for enhanced visualization. The accuracy assessment of the LCP demonstrated an overall accuracy of 83.40% across diverse land cover classes, indicating its efficacy in classification tasks. The plugin's capabilities offer significant potential for applications in land management, environmental surveillance, and urban planning, revolutionizing current practices in land cover classification within the realm of GIS software.

1. Introduction

1.1 Background of the Study

Land cover classification plays a pivotal role in various domains, such as environmental monitoring, urban planning, and ecological assessment, forming the basis for informed decision-making by researchers and policymakers. Typically, land cover maps derived from aerial or satellite imagery, annotated by human experts, serve as indispensable tools for understanding land use dynamics (Richard, 2018).

While contemporary Geographic Information System (GIS) software, exemplified by platforms like ArcGIS (ESRI, 2011), SuperMap (SuperMap, 2016), and eCognition (Trimble, 2014), offers robust image processing and classification capabilities, there are inherent limitations. Notably, these methods are often cost-intensive, time-consuming, and reliant on laborious manual intervention. Users are tasked with meticulously configuring preprocessing parameters, conducting image segmentation, and employing Supervised Classification algorithms for accurate classification (Ulmas, 2020).

The emergence of the Semi-Automatic Classification Plugin (SCP), a product of Luca Congedo, has garnered significant attention within the GIS and remote sensing communities due to its open-source nature and comprehensive toolset tailored for remote sensing data processing (Congedo, 2021). While SCP streamlines various phases of land cover classification, its applicability is confined primarily to Landsat image data and necessitates preprocessing efforts. Moreover, its classification algorithms must catch up to contemporary techniques, failing to harness the potential of modern machine-learning methodologies.

In recent years, the advent of deep learning, epitomized by milestones like AlexNet's (Krizhevsky, 2012) breakthrough in the 2012 ImageNet competition, has revolutionized remote sensing classification methodologies (Lee, 2022). Deep learning techniques, particularly convolutional neural networks (CNNs), have demonstrated unparalleled accuracy in computer vision tasks, offering a promising avenue for automated land cover classification. However, the complexity of implementing deep learning algorithms remains a barrier for many GIS specialists, necessitating a user-friendly solution (Xing, 2022) (Li. 2019).

The current shortcomings of SCP, including labor-intensive processes, time constraints, limited flexibility, and accuracy issues, underscore the need for a novel approach. Addressing these challenges necessitates the development of a new plugin harnessing CNNs, specifically the Pix2Pix architecture (Isola, 2017), to automate the classification process and generate classified static maps efficiently. By leveraging models trained on extensive GIS and remote sensing datasets, this plugin aims to eliminate the need for extensive image preprocessing, offering a seamless end-to-end solution for land cover classification within FOSS GIS software. Chapter 2 will delve into the detailed methodology of plugin development, explaining its potential to revolutionize land cover classification practices.

1.2 Concept of the Study

The Pix2Pix model undertakes image processing, culminating in the creation of a georeferenced digitized image. This resultant image is promptly displayed for preview, allowing users to iteratively repeat the process for multiple images, thus generating a series of static map outputs. These outputs may be combined into a cohesive composite image, offering a comprehensive overview of the designated areas.

https://doi.org/10.5194/isprs-annals-X-5-2024-33-2024

The conceptual framework illustrated in Figure 1 delineates a structured methodology for generating classified map outputs. Starting with selecting a region of interest from a repository of raster data, this process pivots to using a plugin. Herein, users manually delineate and extract the desired area, submitting it as input to a trained Pix2Pix model embedded within the plugin.

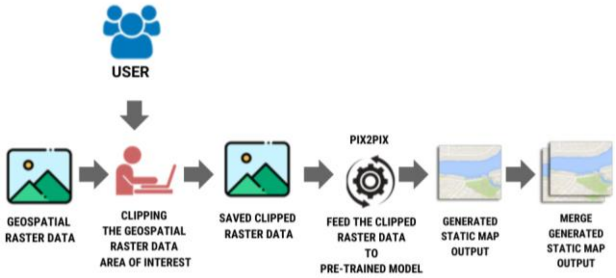

Figure 1. Conceptual Framework

Figure 1 summarizes this systematic process, illustrating the flow from input selection to output generation. In essence, this framework furnishes a methodical approach to furnishing static maps with georeferenced attributes, with potential applications spanning diverse domains such as land management, environmental surveillance, and urban planning.

2. Methodology

2.1 Materials

2.1.1. Software: The project operates within the Microsoft Windows 11, 64-bit environment. Essential software components include GDAL/OGR version 3.6.2 OSGEO, Qt Designer (included within QGIS), PyQt version 5.15.3, PyQGIS (Riverbank Computing, 2021), Visual Studio Code version 1.75.1 as the integrated development environment (IDE), Python version 3.9.5, Pytorch version 1.13.1+cpu, and OpenCV version 4.7. The project is developed and tested using QGIS version 3.28.3-Firenze, 64-bit.

2.1.2. Hardware: The hardware configuration consists of an Intel Core i5-1035G4 CPU @ 1.10GHz, x64 Architecture, paired with 8GB RAM. These specifications suffice for both plugin development and the integration of the pre-trained model.

2.1.3. Datasets: Testing and evaluation utilize raster data from Google Earth, encompassing high-resolution imagery and geospatial information spanning multiple years, including data up to April 2023.

2.2 Overview of the Land Classification Plugin

2.2.1. Land Classification Plugin Architecture: The architectural design of the Land Classification Plugin (LCP), as illustrated in Figure 2, comprises two essential components: the Land Classification Plugin Interface and the Image Segmentation API. Positioned as the Middleware, QGIS (QGIS.org, 2023) serves as the underlying framework for the LCP. QGIS provides users access to various tools, functionalities, and spatial data within this context. Core elements such as the Menu Bar, Toolbars, Viewport, and Plugins constitute the interface, offering a comprehensive platform for users to interact with geographic information.

Central to the LCP's functionality is the QGIS Viewport, a critical component for visualizing geographic data. This component facilitates the selection of Regions of Interest (ROIs) on the map canvas, enabling the plugin to clip and process relevant areas for classification precisely. Once the canvas clipping process is completed, the image data undergoes classification leveraging the PyTorch image segmentation API. Moreover, the LCP leverages the flexibility and power of the Python programming language in conjunction with various APIs and libraries to enhance its capabilities.

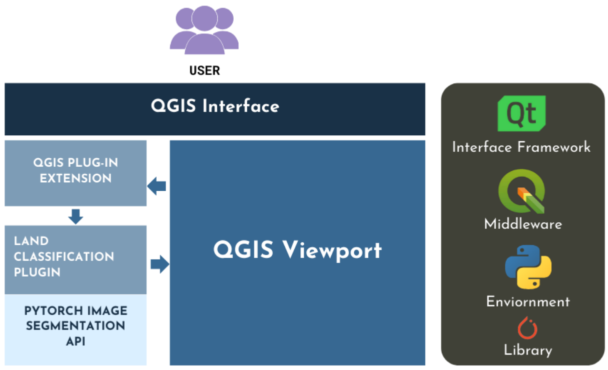

Figure 2: Land Classification Plugin (LCP) Framework

2.2.1 Developing the LCP QGIS Extension: The QGIS has integral tools for developing QGIS plugins, the QGIS Plugin Builder and QT Designer.

The QGIS Plugin Builder (QGIS.org, 2023) is a specialized utility that simplifies the creation of QGIS plugins by generating a structured project framework. This automation minimizes developers' time on manual setup tasks, allowing them to concentrate on implementing plugin logic and features. By enhancing efficiency and maintaining consistency in plugin development workflows, Plugin Builder significantly contributes to developers' productivity within the QGIS ecosystem.

In contrast, Qt Designer (QGIS.org, 2023) is a robust graphical user interface (GUI) design tool within the Qt framework. It empowers developers to design and prototype user interfaces for Qt-based applications, including QGIS plugins. Qt Designer's intuitive drag-and-drop interface facilitates the creation of UI layouts and widget configurations, accommodating developers with varying levels of expertise. Once the UI design is finalized in Qt Designer, developers seamlessly integrate it into their QGIS plugin projects using the generated .ui files within their Python codebase.

While QGIS Plugin Builder automates the setup and structure of QGIS plugin projects, Qt Designer enables developers to design and refine their plugins' graphical user interfaces quickly. Together, these tools streamline the development process for the Land Classification Plugin, ultimately enhancing the functionality and user experience of the QGIS platform.

2.2.2. Land Classification System: The Land Classification System utilizes the PyTorch Image Segmentation API to execute Satellite Image to Classified Maps translation through the Pix2Pix model, as depicted in Figure 3. Employing generative adversarial networks (GANs), the Pix2Pix model is adept at image-to-image translation tasks. Pre-Trained on a comprehensive imagery dataset of orthoimages, the model produces an output akin to an OpenStreetMap (OSM)-like representation, effectively classifying objects within the imagery. This model has only five classes: Water, Buildings, Vegetation, Roads, and Land.

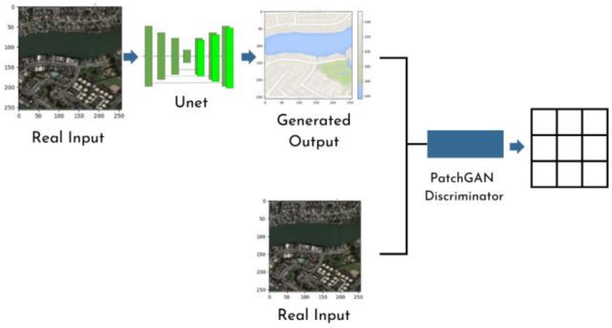

Figure 3. Pix2Pix Architecture

2.2.3. Workflow Process of Land Classification Plugin: The workflow process of the Land Classification Plugin (LCP) shown in Figure 4 begins with thorough data preparation, ensuring the import of accurately georeferenced and aligned remote sensing imagery within QGIS. This foundational step establishes a robust groundwork for subsequent classification. Moving forward, the image classification phase entails the selection of regions of interest (ROIs), which are then processed through the LCP's Clipping and Classification tabs. The plugin autonomously executes classification tasks using the trained pix2pix model, minimizing user intervention. Post-classification, the generated static map outputs can be exported in various formats for further analysis, including examining land cover and land use patterns through tools such as confusion matrices and accuracy metric calculations. In contrast to the LCP's automated approach, the SCP presents a more intricate and customizable workflow catering to diverse project requirements. The choice between the two plugins ultimately depends on the user's preferences and the complexity of the project at hand.

3. Results and Discussions

3.1 Land Classification Plugin Development

3.1.1 Land Classification Plugin Interface: The Land Classification Plugin Interface is the primary interface facilitating interaction between users and the plugin, providing essential tools and features for analysis. Designed with a user-friendly and intuitive layout, the interface ensures accessibility for users of varying skill levels.

3.1.2. Clipping Tab: Users can precisely define and extract regions of interest (ROIs) from input raster data within the Clipping Tab of the Land Classification Plugin interface shown in Figure 5. Users can use the "+" button to draw on the map canvas and specify the desired clipping area flexibly and accurately, as shown in Figure 6. Additionally, the "arrow" button offers an alternative method for clipping directly on the map canvas. Upon defining the area, the Clipping Tab displays the coordinates of the selected region, as shown in Figure 7, allowing users to verify their selection before executing the clipping function by clicking the "Clip" button. This functionality enables users to efficiently extract specific ROIs from large datasets, enhancing the analysis process.

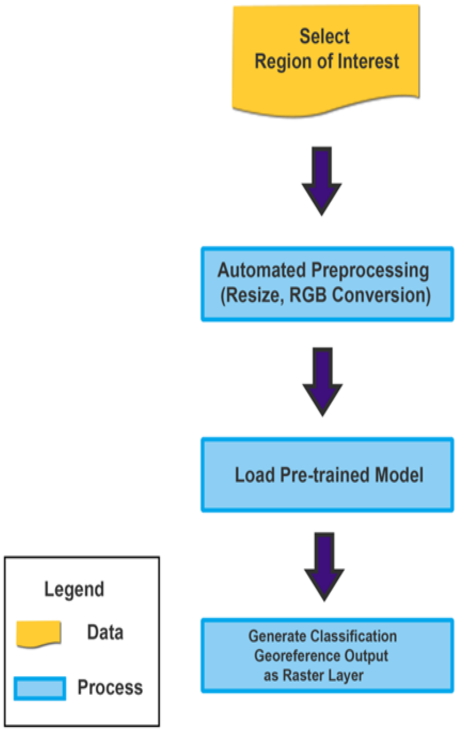

Figure 4. Land Classification Plugin Workflow

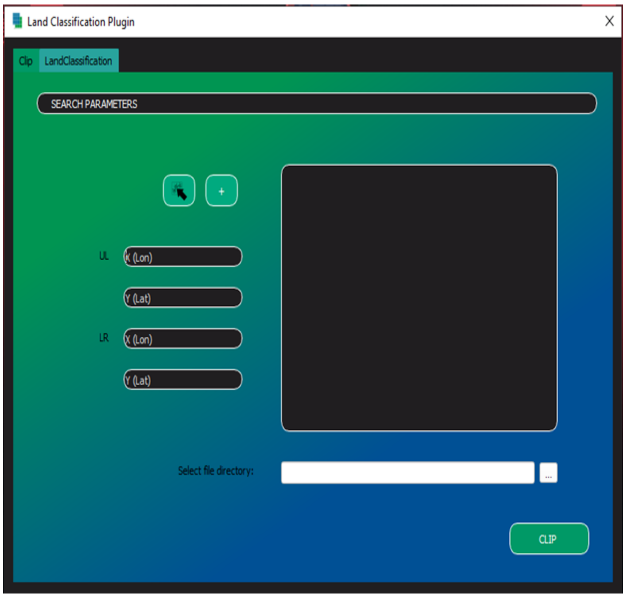

Figure 5. Land Classification Plugin Clipping User Interface.

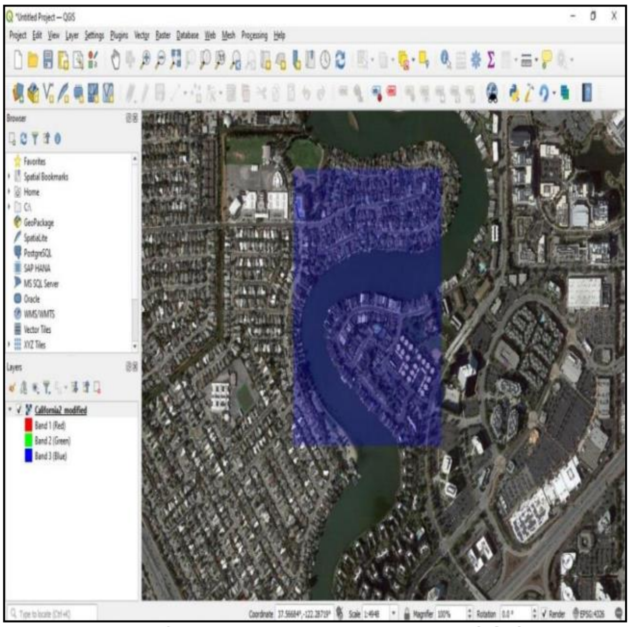

Figure 6. Drawing on a map canvas on QGIS.

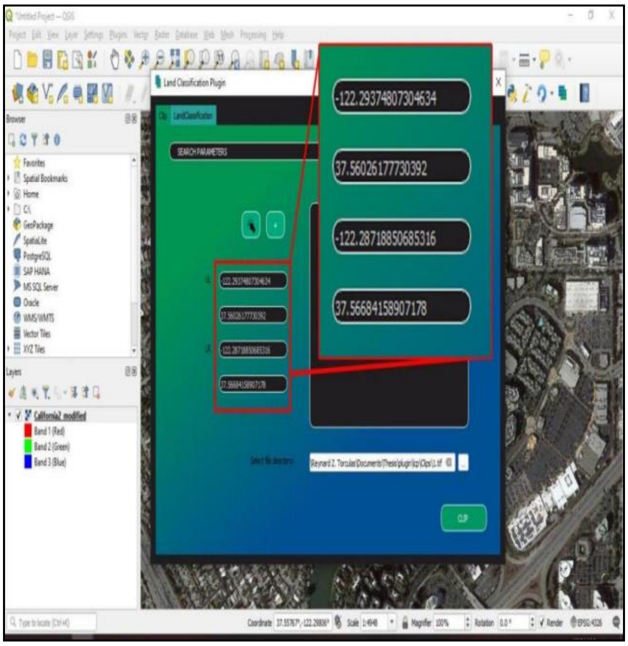

Figure 7. Displaying coordinates of clipped selected area

Following the clipping process, users can preview the output of the clip function within a dedicated window, ensuring the accuracy and appropriateness of the clipped data before further action, as shown in Figure 8. This preview feature allows users to inspect the results and make necessary adjustments, contributing to confidence and efficiency in the analysis process.

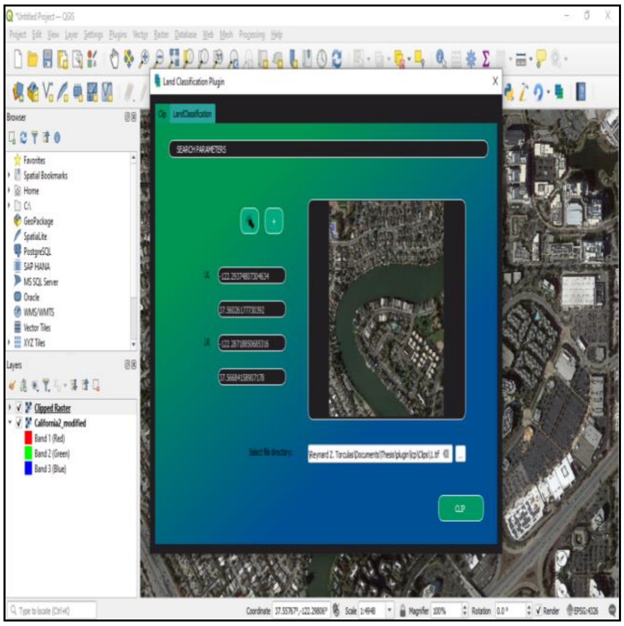

Figure 8. Displaying the output of the Clipped Area

3.1.3. Classification Tab: In Figure 9, users can preview the clipped data for classification purposes by transitioning to the Land Classification Tab User Interface. The Input and Output Preview Windows facilitate the selection and review of clipped files, providing visual representations to aid users in confirming the accuracy and suitability of the input data, as shown in Figure 10.

The land classification process is orchestrated through steps facilitated by the developed land classification plugin. Initiated by the user's clicking the "RUN" button, the plugin initiates the sequence by loading the essential Pix2Pix model into the system, a pivotal component for subsequent operations.

To prepare the input data for processing, the plugin employs the Pillow library, resizing the data to a standardized resolution of 256x256, as shown in Figure 11. This transformation ensures compatibility with the Pix2Pix model, which has been trained on data of this specific resolution. The input data is also converted into a tensor format, normalized, and subjected to other essential operations to optimize it for subsequent stages.

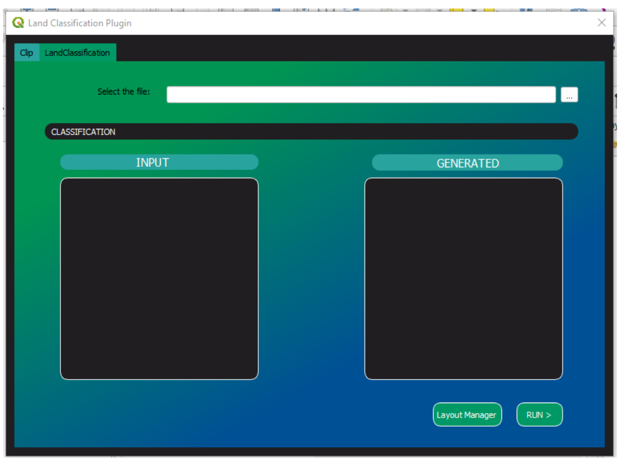

Figure 9. The Land Classification Tab User Interface.

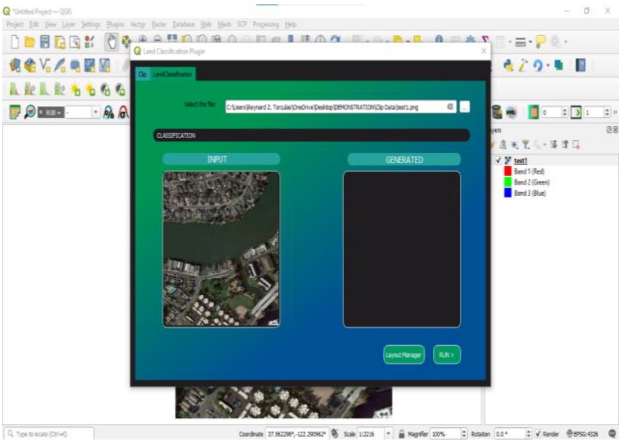

Figure 10. The Input Data Displayed in the Preview Window

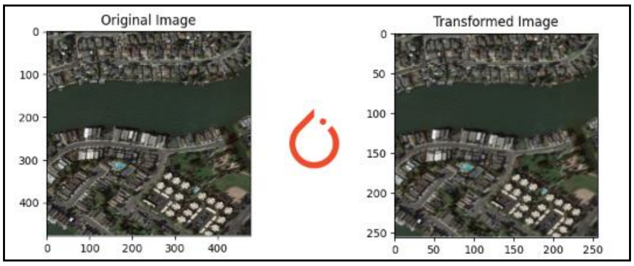

Figure 11. Input data transformed resize into 256x256 resolution.

Once prepared, the input data is fed into the Pix2Pix model, leveraging its trained algorithms to analyze and manipulate it, generating the desired output. The resulting output, in tensor form, undergoes post-processing to enhance its suitability for visualization and analysis. This step involves detaching the output tensor from the computation graph, adjusting its dimensions, clamping values within a specific range, and converting the data to a suitable format.

Following post-processing, the output tensor is converted into a PIL Image object using the Pillow library, facilitating efficient handling and manipulation. Georeferencing the output image is pursued meticulously to align it with spatial references. This involves resizing the output image to match target dimensions, creating a geo-transform using the target file path, and warping the non-georeferenced source image to incorporate necessary spatial referencing data, as shown in Figure 12.

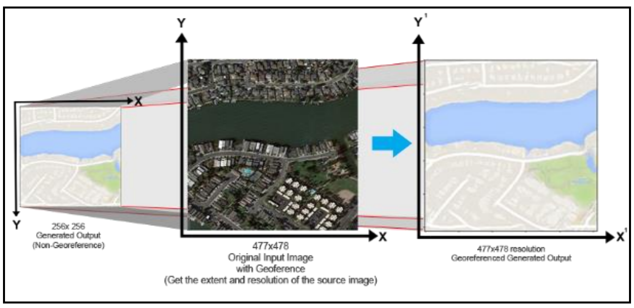

Figure 12. Georeferencing the generated output

Subsequently, the georeferenced output image is converted to JPEG format using the GDAL library, enabling seamless integration into Geographic Information System (GIS) software. Added as a raster layer within the GIS environment, the georeferenced output allows for comprehensive visualization and analysis of land classification results within the spatial context. This rigorous process ensures the effective utilization and compatibility of the generated outputs, facilitating informed decision-making and analysis within the GIS environment.

3.1.4. Layout Manager: Users have the flexibility to engage in the clipping process iteratively, allowing for the creation of multiple static map outputs. Upon selecting "Layout Manager," these individual outputs can be joined into a unified image and incorporated into a pre-designed template utilizing the Layout Manager functionality. This template includes a legend adjacent to the map, facilitating precise classification of depicted features, as shown in Figure 12. By integrating this process, the
The quality of the map is significantly enhanced, fostering improved comprehension of the information presented to users.

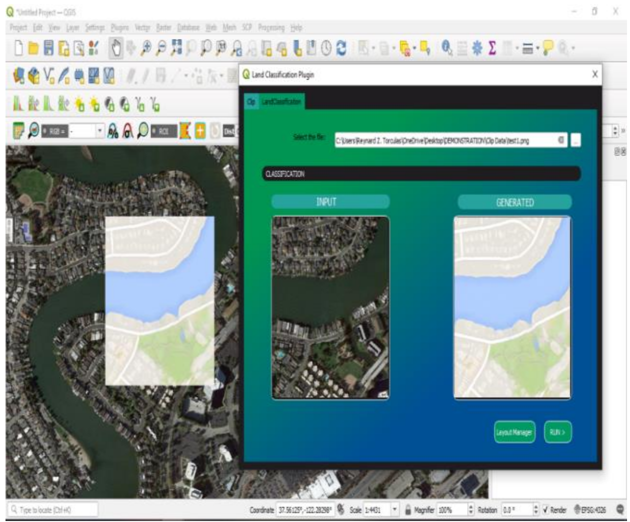

Figure 11. Georeferenced Generated Static Output in QGIS interface and Display the Generated Output in Generated Window in LCP.

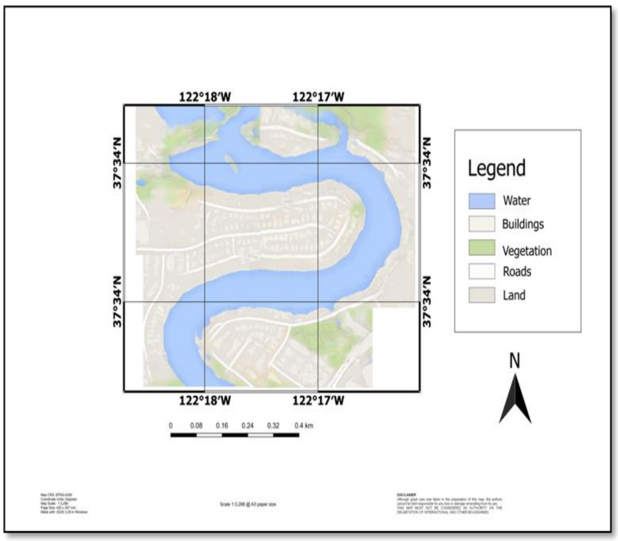

Figure 12. Merged static map output.

3.2 Accuracy Assessment

In assessing the accuracy performance of the plugin, researchers employed three key metrics: overall accuracy, producer's accuracy, and user's accuracy. These metrics are critical benchmarks for evaluating the plugin's efficacy in accurately classifying diverse land cover types.

A comprehensive evaluation was conducted using a curated set of 10 images, each class comprising 10 points, resulting in 50 points per image. This deliberate sampling strategy ensured the acquisition of ample data to thoroughly scrutinize the plugin's performance across a spectrum of land cover types.

3.2.1 Overall Accuracy: Tables 1 and 2 illustrate the confusion matrix, quantifying the proportion of correctly classified pixels across all land cover classes. As depicted in Table 1, the overall accuracy attained by our pre-trained model stands at 83.40%. This result implies that our pre-trained model demonstrates a commendable accuracy in classifying pixels across diverse land cover classes compared to SCP with 69.20%.

		Reference Data					
		Water	Buildings	Vegetation	Road	Land	Total
Classified Data	Water	86	2	3	0	4	95
	Buildings	0	67	2	3	1	73
	Vegetation	0	3	83	0	0	86
	Road	4	2	5	87	1	99
	Land	10	26	7	10	94	147
	Total	100	100	100	100	100	500

Table 1. Confusion Matrix LCP

		Reference Data					
		Water	Buildings	Vegetation	Road	Land	Total
Classified Data	Water	86	0	41	1	3	131
	Buildings	0	86	1	1	11	99
	Vegetation	10	0	57	16	26	109
	Road	0	5	0	68	11	84
	Land	4	9	1	14	49	77
	Total	100	100	100	100	100	500

Table 2. Confusion Matrix for SCP

3.2.2 Producer’s Accuracy: The producer's accuracy is a quantitative measure evaluating the fidelity of pixel classification across distinct land cover categories. As delineated in Tables 3 and 4, the outcomes derived from our pre-trained model exhibit notable Producer accuracy rates across various classes: **LCP:** Water (86%), Buildings (67%), Vegetation (83%), Road (87%), and Land (94%). **SCP:** Water (86%), Buildings (86%), Vegetation (57%), Road (68%), and Land (49%). These findings affirm the model's proficiency in accurately classifying pixels across diverse land cover classifications.

3.2.3 User’s Accuracy: The user's accuracy metric quantifies the model's ability to classify actual pixels within each land cover class correctly. As illustrated in Tables 3 and 4, our model demonstrates commendable User accuracy rates across various categories: **LCP:** Water (91%), Buildings (92%), Vegetation (97%), Roads (88%), and Lands (64%). **SCP:** Water (66%), Buildings (87%), Vegetation (52%), Roads (81%), and Lands (64%). These results underscore the model's capability to precisely discern different land cover types within the input image.

	Producer's Accuracy	User's Accuracy
Water	0.86	0.91
Buildings	0.67	0.92
Vegetation	0.83	0.97
Road	0.87	0.88
Land	0.94	0.64

Table 3. Producer and User Accuracies for LCP.

	Producer's Accuracy	User's Accuracy
Water	0.86	0.66
Buildings	0.86	0.87
Vegetation	0.57	0.52
Road	0.68	0.81
Land	0.49	0.64

Table 4. Producer and User Accuracies for SCP

4. Summary

The precise classification of images and the labor-intensive nature of image processing constitute fundamental challenges within Geographic Information Systems (GIS), critical for numerous GIS applications. This study addresses these challenges by developing a sophisticated Land Classification Plugin to optimize the image classification process. By streamlining data handling and pre-processing tasks, the plugin aims to significantly enhance the accuracy of classified images compared to conventional methods, concurrently reducing the time and human resources required for classification endeavors. Notably, the plugin's versatility in accommodating diverse data formats confers a distinct advantage to potential users.

In developing the Land Classification Plugin plugin comprises two core components: a User Interface and an integrated pre-trained model dedicated to land classification. Backend implementation leveraged callable Python scripts interfaced through PyQt and PyGIS. Furthermore, using the Python programming language and many APIs and libraries contributed to augmenting the plugin's functionality.

Experimental investigation revealed the superiority of Generative Adversarial Networks (GANs), particularly exemplified by the pix2pix model, in efficiently and effectively classifying GIS data compared to conventional region-based algorithms. The seamless integration of the pix2pix model across diverse image datasets showcased its efficiency and ease of use. These empirical insights underscore the potential of the Land Classification Plugin as a pivotal tool for elevating the accuracy and efficiency of image classification tasks within GIS applications. Future research endeavors may further explore the plugin's capabilities across expansive datasets and varied imagery types and assess its advanced features' applicability in broader realms of GIS research.

5. Conclusions

In conclusion, developing and implementing the Land Classification Plugin (LCP) significantly advances land cover classification within Geographic Information System (GIS) software. By harnessing the power of deep learning techniques, particularly the Pix2Pix architecture, the LCP offers an automated solution that streamlines the classification process, overcoming the limitations of traditional methods. The plugin's seamless integration with QGIS provides users with an end-to-end solution for generating classified static maps, revolutionizing current practices in land cover classification.

Through meticulous methodology involving software and hardware configurations and leveraging essential components such as GDAL/OGR, PyTorch, and OpenCV, the LCP demonstrates commendable accuracy, as evidenced by an overall accuracy of 83.40% across diverse land cover classes. This accuracy, coupled with the plugin's user-friendly interface and layout manager feature for enhanced visualization, signifies its potential for land management, environmental surveillance, and urban planning applications.

In essence, the Land Classification Plugin represents a significant step in automating and improving land cover classification processes, offering a comprehensive solution that enhances efficiency and accuracy in GIS-based classification tasks.

References

Congedo, Luca, 2021. Semi-Automatic Classification Plugin: A Python tool for downloading and processing remote sensing images in QGIS. Journal of Open-Source Software, 6(64), 3172, https://doi.org/10.21105/joss.03172

ESRI 2011. ArcGIS Desktop: Release 10. Redlands, CA: Environmental Systems Research Institute.

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. 2017. Image-to-Image Translation with Conditional Adversarial Networks. Computer Vision and Pattern Recognition (cs.CV) arXiv:1611.07004

Krizhevsky, A., Sutskever, I., Hinton, G. 2012. ImageNet Classification with Deep Convolutional Neural Networks, Advances in Neural Information Processing Systems 25, NIPS.

Lee, C. 2022. AlexNet - ImageNet Classification with Deep Convolutional Neural Networks. Noisrucer.

Li, M., Perrier, E., Xu, C. 2019, Deep Hierarchical Graph Convolution for Election Prediction from Geospatial Census Data. The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19), 647-654

QGIS.org., 2023. QGIS Geographic Information System. Open-Source Geospatial Foundation Project. http://qgis.org

Richard, A., Benbihi, A., Pradalier C., Perez V., Couwenberghe, R. Automated segmentation of land use from overhead imagery. International Conference on Precision Agriculture, Jun 2018, Montreal, Canada.

Riverbank Computing. 2021. PyQt. Retrieved February 21, 2023. https://www.riverbankcomputing.com/software/pyqt/

SuperMap, 2016. SuperMap International Limited. 19 Des Voeux Road Central, Central, Hong Kong

Trimble. (2014). eCognition Developer 9.0. Munich, Germany: Wikipublisher

Ulmas, P., Liiv, I. 2020. Segmentation of Satellite Imagery using U-Net Models for Land Cover Classification. ArXiv, abs/2003.02899.

Xing, J., Sieber, R., 2022. The Challenges of Integrating Explainable Artificial Intelligence into GeoAI. Transactions in GIS. Wiley. 626-645.