MapBot: A Multi-Modal Agent for Geospatial Analysis

Demonstration Track

Martin Weiss
Mila, Quebec AI Institute
Montreal, Canada
Polytechnique Montreal
Montreal, Canada
weissmar@mila.quebec

Nasim Rahaman
Mila, Quebec AI Institute
Montreal, Canada
Max Planck Institute for Intelligent Systems
Tübingen, Germany

Chris Pal
Mila, Quebec AI Institute
Montreal, Canada
Polytechnique Montreal
Montreal, Canada
Canada CIFAR AI Chair
Montreal, Canada

ABSTRACT

MapBot is an interactive system to manipulate, analyze, and visualize geospatial data. It combines frontier computer vision models with a large language model running in a Read–Eval–Print Loop (REPL). Users can upload or select aerial or satellite imagery, annotate objects, and query the data using natural language and a point-and-click interface. The LLM agent loop enables the orchestration of Segment Anything and DinoV2, Python code generation and execution, and the display of results in a web interface. This approach lowers the barrier to geospatial analysis for non-experts, enabling rapid annotation and querying of complex data through dialogue that includes map-based interaction.

KEYWORDS

Geospatial Analysis; Multi-Modal Models; Agentic Systems

ACM Reference Format:
Martin Weiss, Nasim Rahaman, and Chris Pal. 2025. MapBot: A Multi-Modal Agent for Geospatial Analysis: Demonstration Track. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 3 pages.

1 INTRODUCTION

MapBot[1] is an agentic system designed to simplify the process of analyzing geospatial imagery for non-experts. Geospatial data is crucial for applications ranging from precision agriculture to urban planning. However, existing GIS tools often require specialized training, limiting their usability for broader audiences. MapBot bridges this gap by integrating a Large Language Model (LLM) with state-of-the-art vision models in an agentic framework.

The core of MapBot is a frontier LLM. The LLM operates in a Read-Eval-Print Loop (REPL) equipped with functions to invoke vision models including Dino V2 and Segment Anything Model (SAM). The LLM generates 'thoughts', writes Python code, and responds to user inputs via chat messages and manipulating the geospatial data rendered on a map. The code generated by the LLM is executed on a Python interpreter which has access to custom functions (implemented using libraries such as GeoPandas [4]) that can manipulate the displayed geospatial data. A 'side effect' of updating the data is that it updates the map displayed to the user. At that point the user may click additional locations on the map and send more chat messages to continue the interaction.

Users interact with MapBot through a web interface built with Streamlit and MapBox [6] which allows them to annotate and query the data directly in their web browser. This interface reduces the barrier to entry since no specialized desktop hardware or software application is required.

Compared to traditional GIS tools that require specialized training and domain-specific software proficiency [2–4], MapBot is designed for non-experts. While some interactive Earth observation systems have begun integrating large language models in agent loops [1], MapBot combines a user interface designed for non-experts with an intuitive self-documenting agent loop and cloud-based neural pipeline. Building on Wu and Osco [11]—which integrated SAM into geospatial workflows — and Osco et al. [10] — which pioneered zero- to one-shot segmentation in remote sensing — MapBot aims to democratize access to such systems by contributing an easy-to-use interface for interactive annotation and analysis.

2 MAPBOT SYSTEM ARCHITECTURE

MapBot uses three state-of-the-art foundation models: GPT-4 [7], Segment Anything Model (SAM) [5], and DinoV2[9], operating together as illustrated in Figure 1, which shows a flowchart diagram of the MapBot architecture. The diagram illustrates how user inputs flow through the LLM to generate Python code, which then interfaces with a REPL to execute geospatial analysis functions. The system shows the bidirectional interaction between user messages/clicks and map updates.

MapBot uses the Python libraries GeoPandas [4] and Rasterio [3] as well as the aforementioned computer vision models to manipulate geospatial data via Python function calls. By using these models and libraries, we can implement highly abstract functions like 'ExtractObjectsThatLookLike', 'FindNearbyObjects', and 'VisualizeObjects'. Furthermore, this collection of function calls is easily extensible to technical specialists who wish to add new functionalities to MapBot (which will be open-sourced upon publication).

MapBot's visual inference pipeline, depicted in Figure 2, proceeds as follows. User clicks and raster imagery serve as inputs. In addition, MapBot can access up-to-date information via external APIs including Mapillary [2], Mapbox [6], and OpenStreetMap [8]. These inputs are processed by neural modules, including the Segment Anything Model (SAM) for segmentation and DinoV2

[1]Demo Video: https://www.youtube.com/watch?v=qNCmsexvrt8

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

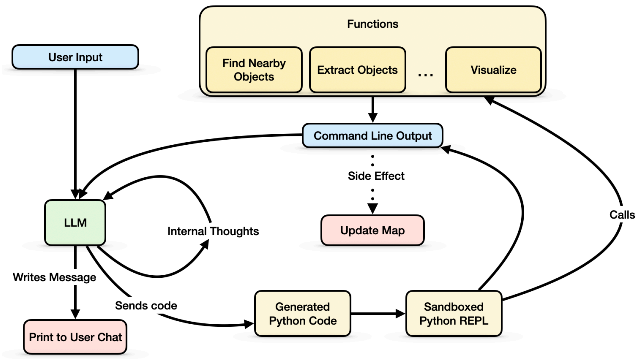

Figure 1: The MapBot system architecture.

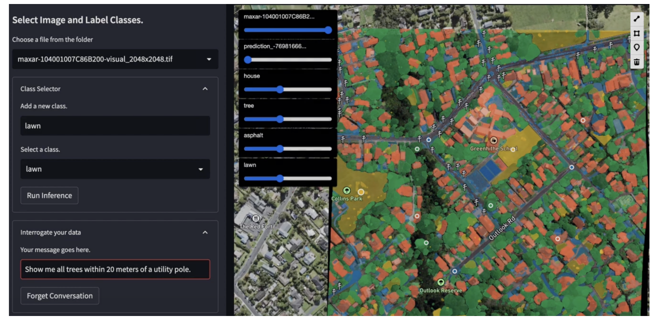

Figure 3: The MapBot User Interface

for feature extraction. The outputs of these models are then fused during an integration phase, resulting in a Pandas GeoDataFrame that merges spatial information with model outputs. This process generates probability maps and segmented outputs.

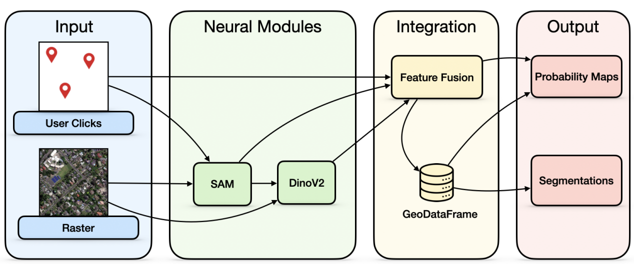

Figure 2: The MapBot neural pipeline flow diagram.

3 USER INTERFACE

The **main components** of the interface include:

(1) **Central Map:** Main geospatial visualization area.
(2) **Sidebar Tools:** Label and visualization controls.
(3) **Class Manager:** Create and manage annotations.
(4) **Annotator:** Point-and-click map labeling tools.
(5) **Inference Panel:** LLM execution controls.
(6) **Layer Options:** Adjust layer visibility and opacity.
(7) **Chat:** Interact using natural language.
(8) **Code Display:** Show LLM reasoning and code.

These interface elements are shown in Figure 3. The user interacts with MapBot via MapBox JS map on a Streamlit website.

4 DEMONSTRATION AND USE CASES

We showcase MapBot's interface and analysis features by walking through a typical user workflow. The process begins when users upload or select a georeferenced raster (GeoTIFF), which then appears on the central map panel alongside sidebar controls.Users can define classes (e.g., "house", "tree") and annotate examples directly on the map. They can then run inference on the ML pipeline and visualize the results on the map. This iterative process allows users to refine the model's understanding by adding classes or annotations as needed.

After annotation, users can run inference on the entire raster, with results displayed as overlays on the map. These include probability maps and class-specific detections, which users can explore by adjusting layer visibility and opacity.

A key feature is the natural language querying capability. The "Interrogate Data" panel allows users to ask complex questions, for example:

- Identify trees near utility poles
- Modify and show certain segmented objects
- Compute statistics about vectorized geospatial data

The demonstration highlights MapBot's application in utility infrastructure management, but its potential extends far beyond. The approach demonstrated by MapBot may benefit numerous fields, including biodiversity sciences, agriculturalists, journalists, and NGOs. By offering advanced geospatial analytics through a user-friendly interface and natural language interaction, MapBot makes powerful spatial data analysis accessible to a wide range of users across various disciplines.

5 CONCLUSION AND FUTURE WORK

MapBot lowers the barrier to geospatial analysis with its agentic orchestration of AI models, intuitive interface, and natural language querying. Future work includes integrating domain-specific plug-ins for environmental monitoring, supporting real-time data streams, and systematically evaluating usability through user studies. In addition, bridging MapBot with large-scale geospatial resources (e.g., Google Earth Engine or Microsoft Planetary Computer) could significantly expand data coverage and streamline data ingestion.

Beyond these core features, the MapBot system architecture naturally accommodates expansion via new functions that could enable it to support specialized data sources or processing routines. For example, researchers focusing on environmental conservation might connect to satellite-based time series data to inject information about changes in forest cover or habitat availability over months or years. Likewise, civil engineers can integrate high-resolution LiDAR datasets for urban planning tasks such as detecting rooftop solar potential or assessing flood risk near critical infrastructure. Developing a plugin 'marketplace' would create new opportunities for users and researchers to apply this approach to diverse geospatial challenges across multiple disciplines.

REFERENCES

[1] Yuxing Chen, Weijie Wang, Sylvain Lobry, and Camille Kurtz. 2024. An LLM Agent for Automatic Geospatial Data Analysis. arXiv:2410.18792 [cs.CY] https://arxiv.org/abs/2410.18792

[2] Omar Ali Christopher Beddow, Saif UI Islam. 2024. Mapillary Python SDK. https://github.com/mapillary/mapillary-python-sdk

[3] Sean Gillies et al. 2013. Rasterio: geospatial raster I/O for Python programmers. https://github.com/mapbox/rasterio

[4] Kelsey Jordahl, Joris Van den Bossche, Martin Fleischmann, Jacob Wasserman, James McBride, Jeffrey Gerard, Jeff Tratner, Matthew Perry, Adrian Garcia Badaracco, Carson Farmer, Geir Arne Hjelle, Alan D. Snow, Micah Cochran, Sean Gillies, Lucas Culbertson, Matt Bartos, Nick Eubank, maxalbert, Aleksey Bilogur, Sergio Rey, Christopher Ren, Dani Arribas-Bel, Leah Wasser, Levi John Wolf, Martin Journois, Joshua Wilson, Adam Greenhall, Chris Holdgraf, Filipe, and François Leblanc. 2024. *geopandas/geopandas: v1.0.1.* https://doi.org/10.5281/zenodo.2585848

[5] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. 2023. Segment Anything. arXiv:2304.02643 [cs.CV] https://arxiv.org/abs/2304.02643

[6] Mapbox. 2024. https://github.com/mapbox/mapbox-gl-js

[7] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774

[8] OpenStreetMap contributors. 2017. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org.

[9] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. DINOv2: Learning Robust Visual Features without Supervision. arXiv:2304.07193 [cs.CV] https://arxiv.org/abs/2304.07193

[10] Lucas Prado Osco, Qiusheng Wu, Eduardo Lopes de Lemos, Wesley Nunes Gonçalves, Ana Paula Marques Ramos, Jonathan Li, and José Marcato. 2023. The Segment Anything Model (SAM) for remote sensing applications: From zero to one shot. *International Journal of Applied Earth Observation and Geoinformation* 124 (2023), 103540. https://doi.org/10.1016/j.jag.2023.103540

[11] Qiusheng Wu and Lucas Prado Osco. 2023. samgeo: A Python package for segmenting geospatial data with the Segment Anything Model (SAM). *Journal of Open Source Software* 8, 89 (2023), 5663. https://doi.org/10.21105/joss.05663