

ЗАСТОСУВАННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ ДО МУЛЬТИМОДАЛЬНОГО АНАЛІЗУ ЗОБРАЖЕНЬ З GOOGLE STREET VIEW

В. Ю. Савчук^{1,а}, Н. І. Недашківська¹

¹ НТУУ «КПІ ім. Ігоря Сікорського», НН ІІСА

Анотація

Розглядається застосування великих мовних моделей для вирішення задачі визначення географічного положення на основі аналізу зображень з Google Street View. Основна ідея полягає в тому, щоб інтегрувати популярні у наш час мовні моделі із модулем аналізу візуального контексту та просторових ознак, доповнюючи їх механізмом точного перетворення географічних координат у пікселі для нанесення на навігаційну мапу та перевірки якості прогнозу за допомогою онлайн-ресурсу GeoGuessr. Прогнози моделей можуть слугувати основою для побудови вузькоспеціалізованої моделі у майбутньому та її навчання методом дистиляції знань.

Ключові слова: велика мовна модель (LLM), Google Street View, аналіз зображень, геолокація, комп'ютерний зір

Вступ

Визначення точного географічного розташування за зображенням є нетривіальною задачею для систем штучного інтелекту з просторовим мисленням. У наш час об'єми інформації у вигляді візуальних даних швидко збільшуються, а тому зростає потреба у технологіях, здатних інтерпретувати навколишнє середовище та локалізувати об'єкти без попереднього знання їхнього розташування. Вирішення такої задачі має практичне значення для навігаційних систем, геоаналітики, журналістських розслідувань і рятувальних операцій.

Останні досягнення у сфері великих мовних моделей відкривають нові можливості для розв'язання такої задачі [1, 2]. Завдяки здатності аналізувати як текстову, так і візуальну інформацію, великі мовні моделі здатні працювати в ролі інтелектуальних агентів, що наближаються до людського рівня розуміння географічного контексту. У цій роботі розглядається підхід до побудови інтелектуального агента, який включає модуль обробки зображень, механізм побудови коректного та вичерпного запиту до LLM та систему конвертації координат у піксельні координати на карті.

У якості бенчмарку використано онлайн-гру GeoGuessr [3], яка пропонує власну метрику визначення точності прогнозованої геолокації відносно правдивої мітки. GeoGuessr – популярна браузерна гра, яка містить широкий набір колекцій із вуличними зображеннями міст та панорамами природних ландшафтів з Google Street View [4].



Рис. 1. Користувацький інтерфейс GeoGuessr

1. Постановка задачі

Завданням даної роботи є розробка інтелектуального агента, який здатний визначати географічні координати на основі зображення з Google Street View без доступу до його метаданих.

Глобальна задача складається з таких підзадач:

- Виділення панорами з користувацького інтерфейсу GeoGuessr (рис. 1), первинна обробка зображення та зведення до формату, придатного для LLM;
- Формулювання вичерпного запиту до однієї чи кількох мовних моделей для визначення географічного розташування місця зйомки;
- Нанесення прогнозу на географічну карту із врахуванням спотворень, спричинених проекцією Меркатора;
- Збереження змісту прогнозу, правдивої мітки та оцінки результатів бенчмарку GeoGuessr до бази даних.

Сформуємо математичну постановку задачі про-

^аsavchuk.v03@gmail.com

гнозуючого модуля інтелектуального агента.

Нехай маємо зображення I з сервісу Google Street View, для якого відсутні метадані про геолокацію. Зображення I розглядається як елемент множини всіх можливих панорам $\mathcal{I}$. Карта мапи світу $\mathcal{M} \subset \mathbb{R}^2$ обмежена прямокутним регіоном з координатами $(\phi_{\min}, \lambda_{\min})$ і $(\phi_{\max}, \lambda_{\max})$, де ϕ – географічна широта, λ – географічна довгота.

Потрібно побудувати функцію географічної локалізації:

$$f : \mathcal{I} \rightarrow \mathcal{M},$$

таку, що $f(I) = (\hat{\phi}, \hat{\lambda})$ позначає прогноз географічної широти і довготи місця зйомки зображення I .

Критерій якості оцінюється за двома метриками:

1. Відстань прогнозу до правдивої позиції в кілометрах:

$$d((\hat{\phi}, \hat{\lambda}), (\phi^*, \lambda^*)) = \text{Haversine}((\hat{\phi}, \hat{\lambda}), (\phi^*, \lambda^*)),$$

де (ϕ^*, λ^*) – істинні географічні координати місця зйомки, *Haversine* – функція обчислення великокурової відстані на сфері.

2. Ігрова метрика GeoGuessr – функція $\text{Score} : \mathbb{R}^+ \rightarrow [0; 5000]$, яка відображає відстань d у кількості балів згідно з формулою:

$$\text{Score}(d) = 5000 \cdot e^{-10 \cdot \frac{d}{d_{\max}}},$$

де d – відстань прогнозу до правдивої позиції, $d_{\max}$ – довжина діагоналі найменшого прямокутника, який містить всі позиції ігрової мапи.

Задача полягає у мінімізації відстані d та максимізації $\text{Score}(d)$ для всіх вхідних зображень.

2. Концепція функціонування інтелектуального агента

Ключову роль у функціонуванні майбутнього інтелектуального агента відіграє взаємодія з однією чи кількома великими мовними моделями. Окрім взаємодії з LLM, агент повинен виконувати набір додаткових дій, пов'язаних із обробкою вхідних та вихідних даних, а також оцінкою прогнозів та їх збереженням до бази даних.

2.1. Великі мовні моделі

Великі мовні моделі – це нейронні архітектури, які здатні генерувати, аналізувати та узагальнювати текстову інформацію на основі великих обсягів навчальних даних [1, 2]. Завдяки інтеграції з мультимодальними механізмами, сучасні мовні моделі здатні також обробляти зображення – екстрагувати з них текстову, контекстну та просторову інформацію. Це розширює сферу їх застосування, дозволяючи вирішувати задачі, які раніше вимагали окремих моделей.

Більшість передових LLM, зокрема OpenAI, Google Gemini та Claude від Anthropic, мають публічні API, що дозволяє інтегрувати їх у власні програмні системи. У рамках цієї роботи для аналізу вхідних зображень, формування гіпотез щодо місцезнаходження та логічного обґрунтування відповідей були використані моделі OpenAI gpt-4o, Google Gemini 2.0-flash та Claude версії 3-7-sonnet.

2.2. Взаємодія з LLM та допоміжними сервісами

Схема взаємодії агента з LLM та допоміжними складовими зображена на рисунку 2.

Взаємодія користувача з агентом починається з калібрування ключових точок екрану та надання їх агентові. Оскільки агент напряду взаємодіє з графічним інтерфейсом браузеру, калібрування є важливим етапом, який істотно впливатиме на кінцеві показники агента.

Ключові точки екрану є параметрами роботи агента, які визначають області розташування панорами та навігаційної карти. Ці області необхідні для виконання скріншоту зображення Google Street View та надання остаточного прогнозу відповідно.

Агент взаємодіє з однією чи кількома мовними моделями, надаючи їм текстовий запит із вказівками щодо акцентів аналізу та зображення, закодоване у форматі base64 для зручної та швидкої передачі мережею. У відповідь агент очікує прогноз у форматі, який був заданий у запиті, із детальною аргументацією, можливими локаціями та географічними координатами найбільш імовірної геопозиції місця зйомки. У випадку, якщо агент реалізує взаємодію з кількома LLM, необхідна реалізація механізму агрегації геопозицій, наданих кожною з моделей.

Провівши конвертацію (параметризовану ключовими точками калібрування) геопозиції в координати на мапі, агент повинен нанести остаточний прогноз на навігаційну карту GeoGuessr.

Отримавши значення критеріїв якості з GeoGuessr API, останньою задачею агента є збереження прогнозів, наданих кожною з моделей, істинних координат місцезнаходження та показників метрик якості в базу даних.

Агент на рисунку 2 представлений єдиним блоком, хоча за своєю внутрішньою структурою він містить окремі функціональні модулі, які будуть розглянуті у розділі 3.

2.3. Характерні ознаки зображення

Коректне формування запиту до великих мовних моделей напряду впливає на якість вихідних прогнозів. Для того, аби вдало провести аналіз зображення, моделі повинні отримати чіткі вказівки щодо характерних ознак зображення, які потрібно враховувати, а також стосовно очікуваного формату відповіді.

Основним кроком при визначенні географічного розташування є визначення набору імовірних країн

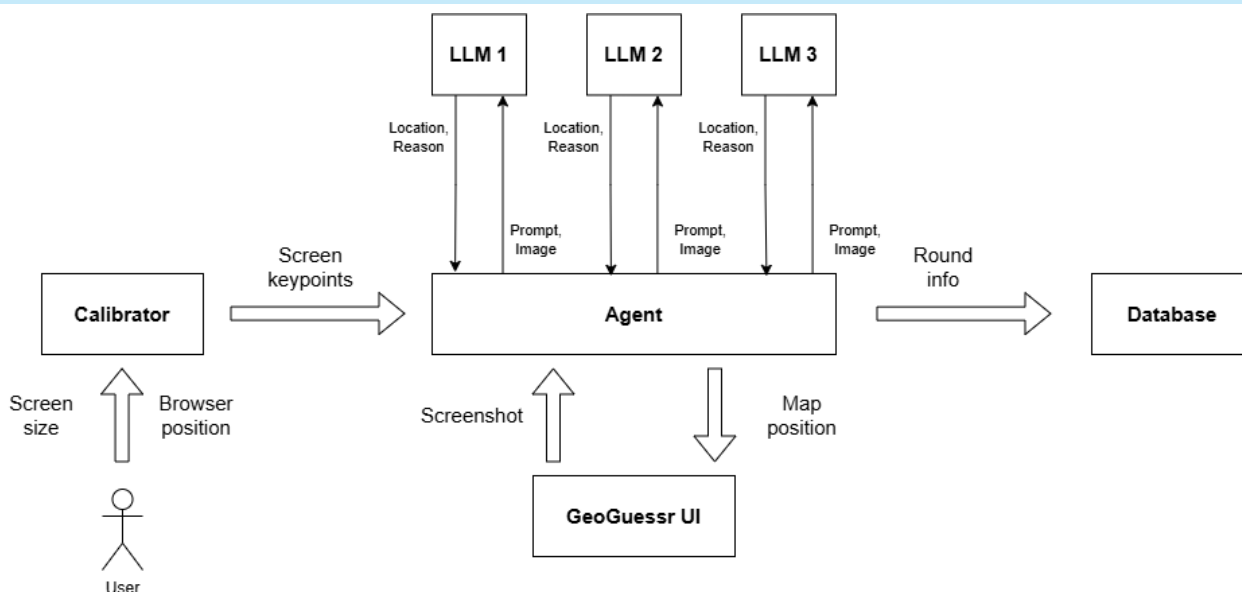


Рис. 2. Загальна схема взаємодії агента з LLM та допоміжними складовими

з огляду на ключові риси навколишнього середовища, які характерні окремим географічним регіонам. Моделі повинні враховувати такі характерні ознаки: стиль та матеріали архітектурних споруд, особливості дорожніх знаків та розмітки, напрямок дорожнього руху, формат номерних знаків на авто, мова тексту на рекламних оголошеннях, особливості флори та клімату. У результаті інтерпретації таких ознак у географічному контексті в багатьох випадках простір можливих варіантів вибору можна звузити до кількох країн або навіть міст чи адміністративних одиниць однієї країни.

3. Програмна реалізація

Цільовий інтелектуальний агент представляє собою багаторівневу систему, яка складається з програмних модулів, кожний з яких вирішує окрему задачу. Для програмної реалізації концепції була обрана мова Python [5], оскільки вона надає зручні інструменти для ефективного розв'язку поставлених на попередньому етапі підзадач (обробка зображень, взаємодія з графічним інтерфейсом, LLM та базою даних).

3.1. Зміст та композиція класів

Окремою задачею, згідно зі схемою на рисунку 2, є задача калібрування ключових точок екрану користувача. Відповідальним за це є клас Calibrator, основна задача якого – сформувати конфігураційний файл keypoints.json із координатами пікселів ключових точок на екрані користувача.

Центральним класом у програмній реалізації є GeoAgent. Його задача – узгодження та координація роботи інших класів. У конструктор класу GeoAgent приймає ключові точки, отримані в результаті калібрування. Розміщення прогнозу на карті делегується класу UIInteractor, який взаємодіє з курсором та

клавіатурою користувача за допомогою бібліотеки pyautogui.

Клас Converter відповідальний за конвертацію географічних координат в координати на карті із використанням проекції Меркатора [6]:

$$y_{\text{Mercator}} = R \cdot \ln \left(\tan(\pi/4 + \phi/2) \right),$$

де y_{Mercator} – вертикальна координата в проекції Меркатора, R – радіус Землі, $\phi \in (-\pi/2; \pi/2)$ – географічна широта в радіанах.

Аргументом логарифму є центральна функція Меркатора.

Таке перетворення застосовується для побудови карт, де координата в проекції Меркатора у має рівномірну шкалу незалежно від широти. Проте при такому підході сильно спотворюються високоширотні області (наприклад, Гренландія та Канада виглядають на картах із такою проекцією значно більшими за свої реальні розміри).

Клас ImageHandler був розроблений для первинної обробки та конвертації зображень у формат base64. Під час обробки, зокрема, відбувається підвищення контрасту та різкості зображення засобами бібліотеки cv.

Для взаємодії з GeoGuessr API створено клас GeoGuessrClient, який дозволяє отримати інформацію про гру за її токеном. У свою чергу токен надає клас BrowserInteractor, який взаємодіє з відкритою вкладкою браузера користувача.

Взаємодію з базою даних клас GeoAgent проводить, використовуючи екземпляри класів-репозиторіїв, кожен з яких створений для роботи з окремою таблицею бази даних.

3.2. Структура бази даних

Збереження прогнозів моделей, оцінок якості прогнозів та інформації про істинні геомітки відбувається завдяки інтеграції агента з MS SQL Server та

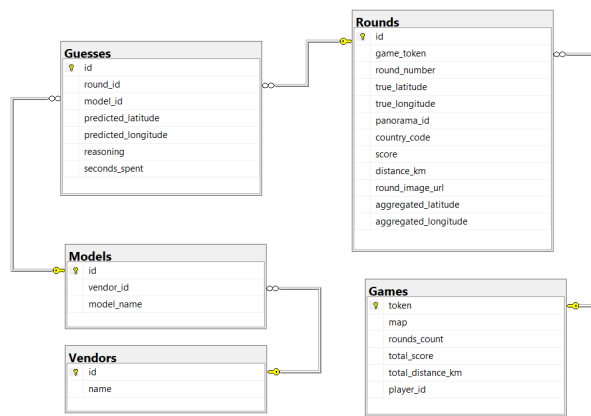


Рис. 3. Структура бази даних

Azure Blob Storage [7] для збереження скріншотів панорам Google Street View. Структура бази даних наведена на рисунку 3.

Таблиці Vendors та Models містять назви платформ та моделей відповідно. У таблиці Guesses зібрані дані про прогнози окремих моделей, витрачений час та аргументацію. Таблиця Game містить інформацію про проведені ігри: назва мапи, токен та сукупні метрики за всіма раундами гри. Таблиця Rounds характеризує окремі раунди гри – істинні мітки та агреговані прогнози моделей, значення метрик та посилання на зображення, яке подається на вхід моделям. Саме зображення зберігається у Azure Blob Storage – хмарному сервісі від Azure, який дозволяє зручно зберігати зображення та інші формати файлів.

Мета формування бази даних – збір інформації для майбутнього навчання вузькоспеціалізованої моделі методом дистиляції знань, тобто на основі прогнозів інших моделей, у даному випадку прогнозів на основі LLM.

Висновки

У результаті виконання даної роботи було розроблено інтелектуального агента із інтеграцією вели-

ких мовних моделей для вирішення задачі географічної локалізації за зображенням.

Було реалізовано повний цикл обробки даних – від вхідної панорами Google Street View на екрані користувача до вихідних географічних координат та оцінки точності за допомогою інструментів GeoGuessr. Завдяки використанню публічних API великих мовних моделей від OpenAI, Gemini та Anthropic, вдалося побудувати агентну систему, яка імітує міркування людини при визначенні місцезнаходження на основі географічного контексту зображення.

Отримані результати свідчать про потенціал застосування LLM у задачах геопросторової аналітики та відкривають перспективи для створення вузькоспеціалізованих моделей локалізації шляхом дистиляції знань із мовних моделей.

Перелік використаних джерел

1. *Auffarth B.* Generative AI with LangChain: Build large language model (LLM) apps with Python, ChatGPT, and other LLMs. — Birmingham: Packt Publishing Ltd, 2024. — 548 с.
2. *Berryman J., Ziegler A.* Prompt Engineering for LLMs: The Art and Science of Building Large Language Model-Based Applications. — Gravenstein Highway North, Sebastopol, CA: O'Reilly Media, Inc., 2024. — 280 с.
3. Plonk it. Beginner's Guide to Geoguessr. — 26.05.2025.
4. GeoTips. Tips and tricks for GeoGuessr game. — 26.05.2025.
5. *Solem J. E.* Programming Computer Vision with Python. — Gravenstein Highway North, Sebastopol, CA: O'Reilly Media, Inc., 2012. — 300 с.
6. *Borovik A., Khudaverdyan H.* Mercator's projection, logarithm and seafaring. — Manchester Institute for Mathematical Sciences. School of Mathematics, 2010. — 27 с.
7. Microsoft learn. Introduction to Azure Blob Storage. — 26.05.2025.