# GeoGuesser AI Agent

LUMEN Data Science Use Case

## 1 About us

Photomath d.o.o.
Strojarska 20, 14th floor

Contact Personnel:

- Marijan Smetko
  - marijan.smetko@photomath.com
- Ivan Jurin
  - ivan.jurin@photomath.com

Photomath is a fast-growing EdTech company whose mobile app is the #1 app in the world to learn math. Powered by advanced machine learning technology, the app instantly scans, accurately solves, and intuitively explains printed and handwritten math problems to users through step-by-step explanations. With over 250 million downloads globally and tens of millions of users worldwide every month, Photomath is the most popular mobile application from Croatia and one of the most popular educational apps of all time. Photomath's mission is to democratize education through the use of artificial intelligence and software-generated educational content. The Photomath team in Zagreb has grown to over 150 employees and more than 130 students working on content creation, and they also have a team of a dozen people in Silicon Valley, USA.

# 2 Challenge Use-Case

Our challenge use case is called GeoGuesser AI Agent, an AI agent which tries to guess the location of the input image.

# 3 Use Case Description

Photomath employees tend to work hard bringing new features and polishing old ones of our math-solving mobile application. However, all work and no play make Jack a dull boy, so we have a fair share of stress relieving methods. One of the most favorite methods of the AI team is a web-based geographic discovery game called GeoGuesser.

Simply put, GeoGuesser throws a player on a random point on the globe, and the player should figure out the location using nothing more than the visual cues from the Google Street View and a compass. This simple game brings hours of fun on a monthly basis by allowing human players to analyze the features of the images (such as vegetation, road driving side, the color of the road lines, language, etc) which might help a person conclude with (in)certainty what the location is. The question we keep asking ourselves is whether or not an AI agent can be trained to perform well on the GeoGuesser, and that's exactly what you are to find out!

## Educational Aspects

This task is unique for its gaming element, while other problems are usually business-oriented. This may come as a relief for the students who are not really apt in business and economics topics. However, the task does not seem too easy; it certainly isn't one of the typical tasks one may come across in the areas of artificial intelligence and computer vision. We believe that is a good thing since it motivates the creativity of the students in coming up with the possible approaches for the solutions. Students will have a chance to get familiarized with the data preprocessing in the computer vision, and also the model definition, training, deployment, and evaluation steps. After the competition ends, they are left with a project which will definitely enrich their CV.

# 4 Tools

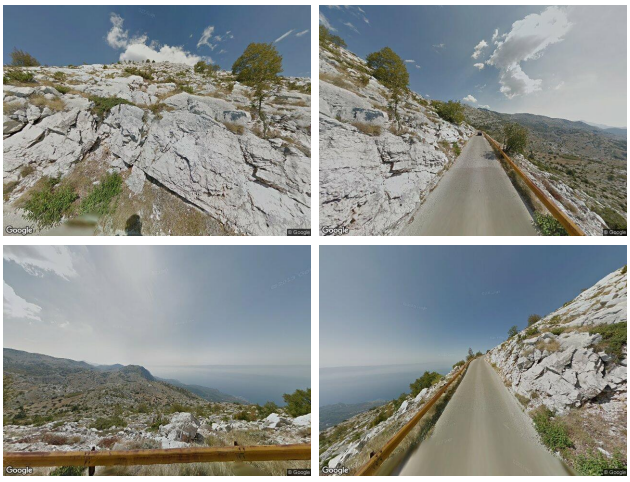
Our recommended programming language is Python. If you plan to use deep learning approaches, we recommend the PyTorch framework. However, student teams are free to use any other frameworks (Tensorflow, MXNet, Caffe2, JAX) or languages (such as C++, Julia, Java, or even Matlab). No extra nor penalty points will be given on the basis of the technology used - we simply don't care; what we really care about is the final product.

# 5 Dataset Description

The dataset will consist of 64,000 images of 16,000 different locations across continental Croatia (this means no islands; however the Dubrovnik-Neretva county, while disconnected from Croatia proper, will be included in the dataset). The output of the AI agent should be a pair of two coordinates: the latitude and the longitude.

The primary metric used for the evaluation of the model's output is the average great-circle distance between the output coordinates and the target coordinates - the smaller the average distance, the more points are awarded to the competing teams.

## Dataset Example

| Input images | Target |
| --- | --- |
|  | 45.3676458 N<br>18.3509275 E |
|  | 45.0542248 N<br>15.1978817 E |
|  | 43.281859 N<br>17.0924535 E |

# 7 Solution Format

At a minimum, the teams should build a REST API that accepts an image (or multiple images) and returns the latitude and the longitude of the predicted location of the input image(s). The design of the API and its endpoints is purely arbitrary, however, it must be documented properly. We will test all the APIs and will grade the ease of setup and usage - more work for us, fewer points you receive. :))

However, extra points will be awarded for the teams that also build a simple web or mobile app which allows a user to upload an image and receive the coordinates back. Extra features such as showing that coordinate on a map of Croatia or anything else you can come up with will also be graded. Both the API and the web app should be well, yet concisely documented - we believe you need no more than 10 pages to achieve this. No penalty for this, though.

Separate from the technical documentation is the project documentation which follows the CRISP-DM methodology and includes the following chapters: business understanding, dataset understanding, data preparation, modelling and evaluation.[1] The project documentation should include more than enough visualizations and infographics to properly convey the data analysis, problems and choices made.

The solution is delivered in two ways:

1. On the last day of the competition, Photomath will release a test set with 16,000 images and the teams should provide the predicted results for that test dataset. This will be evaluated with an average great-circle distance which will count towards the final ranking.
2. One person from each team will upload a zip archive containing:
    - The source code files for model training and evaluation, REST API, and optionally for the web application
        - Alternatively, we accept Jupyter notebooks
    - The model file itself
    - A pdf file containing the project documentation
    - Another pdf file containing the technical documentation

Top graded teams will be invited to the finals to hold a 10 min presentation. The final winners will be determined by the sum of the presentation score and their solution quality.

[1] The Deployment chapter should be handled by the technical documentation.

# 8 Grading Criteria

| Criterion | Grade | Criterion weight | Note |
|---|---|---|---|
| Project documentation | 0-100 | 20% | We grade what we see: the teams should show a high level of problem understanding in a clear manner. Do not spare visuals, examples, and charts. |
| Technical documentation | 0-100 | 20% | We grade correctness, brevity, and clear descriptions of technical issues and design choices. |
| Model accuracy on the test set[2] | 0-100 | 15% | Of course we like better models, however, we like problem-solving approaches and descriptions more. |
| Overall solution quality | 0-100 | 10% | High-level impression about the solution approach, creativity of the solution, solution composition, and real-world usability. |
| Solution feasibility: deployment | 0-100 | 5% | How easy/hard it is to deploy the solution in production. |
| Solution feasibility: extensibility | 0-100 | 5% | How easy/hard it is to add capabilities to the system. |
| Presentation: slides | 0-100 | 5% | How easy/hard it is to understand the slides and the graphs. |
| Presentation: skills & performance | 0-100 | 20% | The storytelling, clarity, examples, analogies and humor all help convey the information, which is actually the most important skill for a data scientist |
| TOTAL | 0-100 | 100% | |
| Extra: web or mobile app or any other extra effort | 0-10 | | The grade depends on the user experience with the app - the emphasis is on the functionality, not design. |

[2] The top performing model will get all the points. Other models' performance will be scaled on a uniform range of points. This means the better your team's model is, the fewer points left for the other teams

# 9 Additional Information

ML Engineer Resources
Geographic coordinate system
Explanation of the coordinate system