LUMEN Data Science

2022.

GeoGuesser AI Agent

Web application documentation

Submission date: *May* 8, 2022

1. Use case

App usage is pretty intuitive, user can upload one or more images on home page either by browsing or drag'n'drop. User can input real coordinates if he wants to check how good can model predict. After loading images by clicking submit button images are sent to the server and feed into the model. Server returns HTML rendered with pinpointed predicted location. If user provided true coordinates also true location will be pinpointed alongside with the line between the two points.

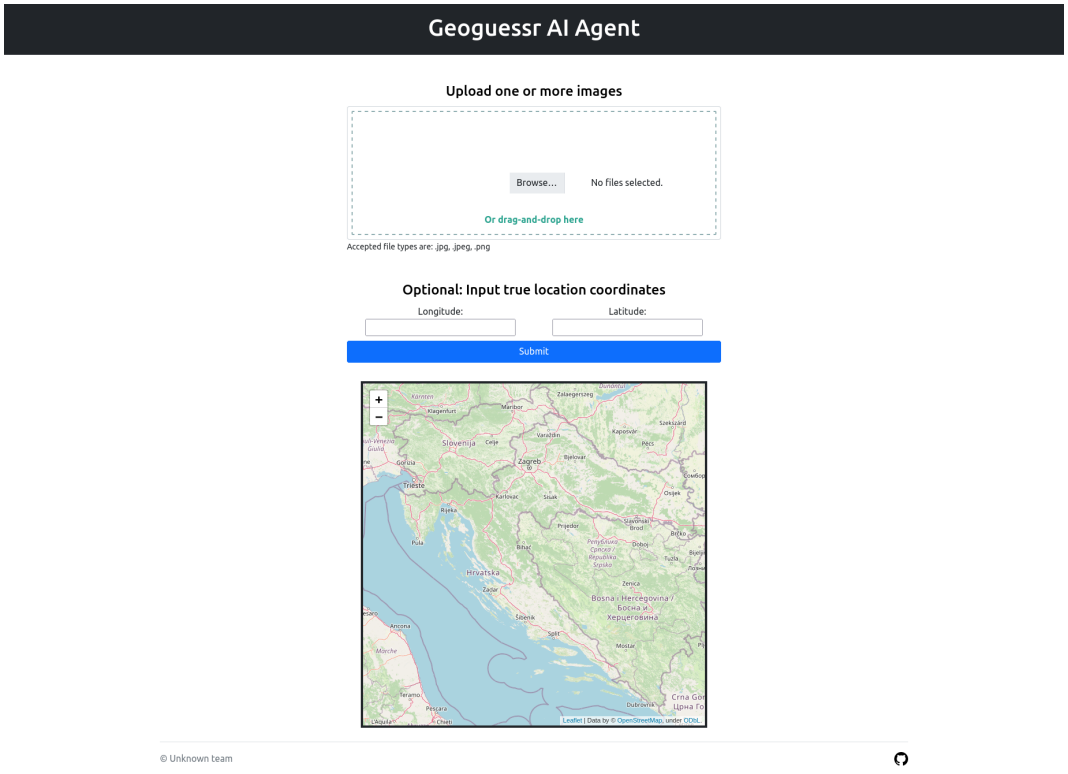

Figure 1.1: Before submission

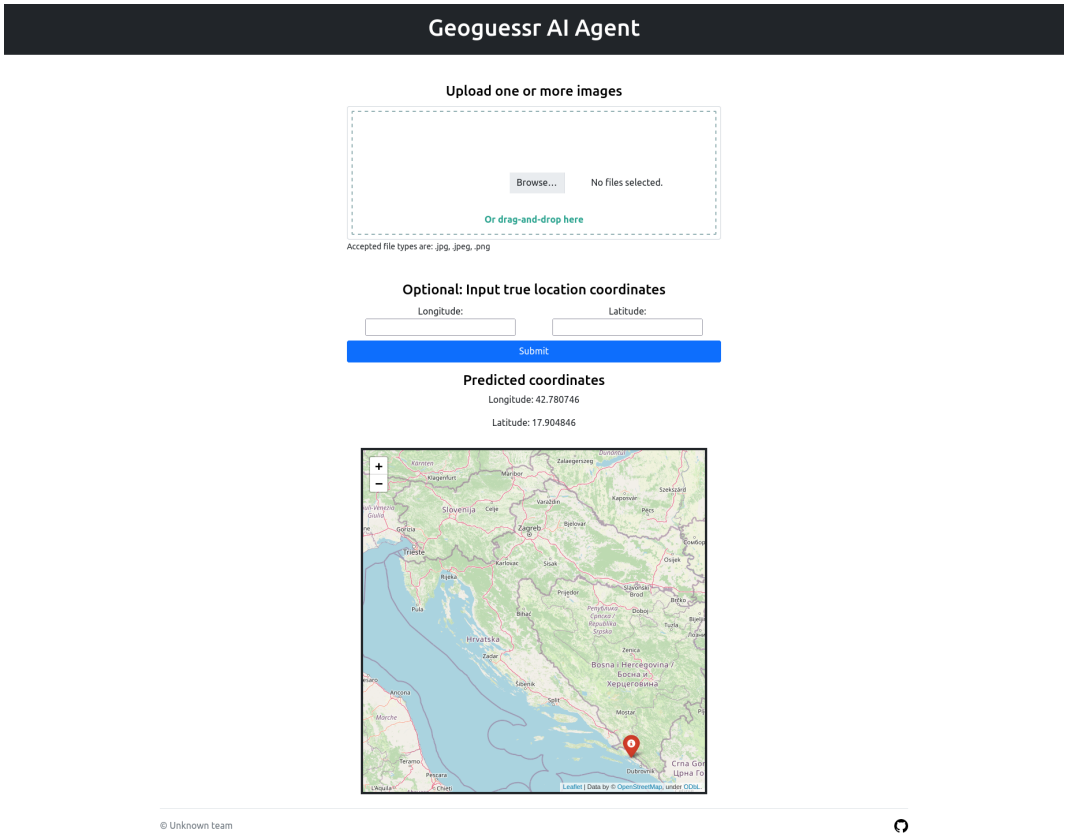

Figure 1.2: After submission(only images)

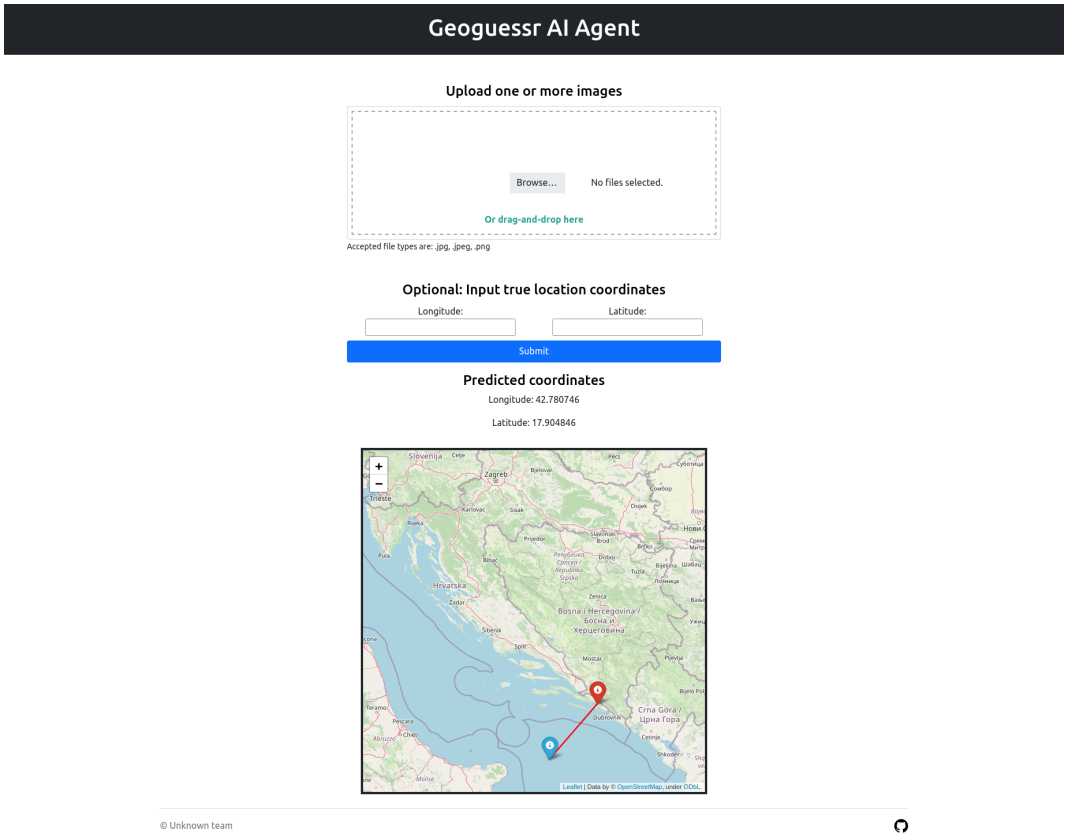

Figure 1.3: After submission(images and real values)

2. Submitting images

Main logic of the whole app is based in the /predict route. Route arguments:

Name	Type	Optional	Description
Images	Array	False	Array of images used for predicting location
Longitude	Float	True	Longitude of the true location
Latitude	Float	True	Latitude of the true location

Image will be converted if it is in convertible formats(PNG,JPEG). After conversion (if needed) image will be resized on the shape suitable for the model input of dimensions (260,260) and prepared for the model input. After required preprocessing image is loaded in the model by calling predict method. Predict method returns image group predictions from the ensemble of models. ensemble predictions are aggregate with mean and passed to the the get_weights method to retrieve weights for every group. Group weights are then passed to calculate_predict_coordinates function which takes care of calculating final ensemble prediction. Method make_html is called with predicted coordinates and true coordinates if they were passed over HTTP request. make_html creates HTML of the map with pinpointed locations. Main responsibility of the function is creating map using python library folium. Function creates map by creating object of the custom wrapper class Visualizer which implements creating map of Croatia, locations points and line between locations. After creating map with points it's stored in the local memory in HTML format.