GEO Intelligence

March 22, 2026

Geo intelligence, particularly Geospatial Intelligence (GEOINT), involves the an alysis of geospatial data to gain insights about objects and events in relation to space and time.

We set the focus on 4 – 5 APIs to extract information from maps or pictures. Geospatial Artificial Intelligence (GeoAI), as the integration of geospatial studies and AI, has become one of the fastest-developing research directions in spatial data science and geography.

1. Object Detection API
2. Image to Text API
3. Language Translator
4. Country, Capital API
5. Image Random Generator (to train data)

Lets take a look at the following scene:

Test and Train picture of a standard normalized scene

You see houses, cars, a fence, maybe something like a human being, you see clouds and trees and also an unidentified object bottom right behind the fence

(mabe a bench). Also hidden information (as a white watermark) is behind the fence from "© 2025 Google" or a white and red marked torch, could be a train barrier. So the quesion is: Where it is in the meaning of the location? A first approach could be a similarity search or a simple object detection, but the challenge is growing.

The second are correlated objects. True intelligence can be the ability to confirm the probability of one object by a second, correlated object. In the example of the fence above, there are usually cars in front of the fence; so the probabilty of a fence gains in addition!

This rapid change in the field calls for a deeper understanding of the recent developments and envision where the field is going in the near future. So lets practice with the first API. The Object Detection API provides fast and accurate image object recognition using advanced neural networks developed by machine learning experts. Given an input image, return a list of detected objects labels, confidence percentages and bounding boxes. Objects with confidence less than 0.3 (30%) are filtered out.

```
function PyCodeGeoJSON_ObjectDetect(imgpath, aAPIKey: string)
begin
  with TPythonEngine.Create(Nil) do begin
  //pythonhome:= 'C:\Users\User\AppData\Local\Programs\Python
  try
    loadDLL;
    autofinalize:= false;
    ExecString('import requests, geojson');
    //execstr('point = geojson.Point((8.5417, 47.3769))  # (l
    ExecStr('url= "https://api.api-ninjas.com/v1/objectdetect
    ExecStr('image_file_descriptor = open("'+imgpath+'","rb")
    ExecStr('headers= {"X-Api-Key":"'+aAPIKey+'"}');
    ExecStr('files = {"image": image_file_descriptor}  ');
    ExecStr('r = requests.post(url, headers=headers, files=fi
    result:= (EvalStr('r.json()'));
  except
    raiseError;
  finally
    Free;
  end;
 end;
end;
```

Real Python Template:

```
import requests

api_url = 'https://api.api-ninjas.com/v1/objectdetection'
image_file_descriptor = open('YOUR_IMAGE.jpeg', 'rb')
files = {'image': image_file_descriptor}
r = requests.post(api_url, files=files, headers={'X-Api-Key':
print(r.json())
```

A JSON array of objects, each representing a detected object with its label, confidence score, and bounding box coordinates, or an error if the request is unsuccessful. It finds 32 objects like this:

```
[{'label': 'car', 'confidence': '0.78', 'bounding_box': {'x1':
'253', 'y1': '455', 'x2': '458', 'y2': '540'}},{'label':
'fence',
```


You can play around this with my script or a live demo online:

Object Detection API – API Ninjas

Download 1473_Geo_Intelligence_API1_4.txt (maXbox5)

https://sourceforge.net/projects/maxbox5/files/examples/1473_Geo_Intelligence_API1_4.txt/download

The next step is to extract text, chars or signs from the fetched landscape.
The Image to Text API detects and extracts text from images using state-of-the-art optical
character recognition (OCR) algorithms. It can detect texts of different sizes, fonts, and even handwriting.

```
function PyCode2imagetotext2(imgpath, aAPIKey: string): strin
begin
  with TPythonEngine.Create(Nil) do begin
  try
    loadDLL;
    autofinalize:= false;
    ExecString('import requests');
    ExecStr('api_url= "https://api.api-ninjas.com/v1/imagetot
    //ExecStr('image_file_descriptor = open(r'+imgpath+', "rb
    ExecStr('image_file_descriptor = open(r"'+imgpath+'", "rb
            'files = {"image": image_file_descriptor}'+lf+
            'headers= {"X-Api-Key":"'+aAPIKey+'"}'+lf+
            'r = requests.post(api_url, headers=headers, file
    result:= ((EvalStr('r.json()')));
  except
    raiseError;
  finally
    free;
  end;
 end;
end;
```

When you got the error debug: 95-
SyntaxError: unexpected character after line continuation character (line 1, offset 33): 'image_file_descriptor = open(r.\examples\geocitytextrandimage0221.jpg, "rb")' 937 err:20 then you
change ExecStr('image_file_descriptor = open(r'+imgpath+', "rb")'); to
ExecStr('image_file_descriptor = open(r"'+imgpath+'", "rb")');

The result can be:

```
[["[{'text': 'Supertramp', 'bounding_box': {'x1': 148, 'y1': 26,
'x2': 262, 'y2': 46}}, {'text': '偶数', 'bounding_box': {'x1':
285, 'y1': 26, 'x2': 334, 'y2': 42}},
```

Then in a process pipeline 1-4 we translate the extracted text in for example japanese (like above):

```
function Text_to_translate_API54(AURL,aclient,lngorig,lngtarg
var httpq: THttpRequestC; //THttpConnectionWinInet;
    jo: TJSON; jarr:TJsonArray2; urlback: string;
  begin
    httpq:= THttpRequestC.create(self);
    httpq.headers.add('Accept: application/json; charset=utf-
    //httpq.SecurityOptions:= [soSsl3,soPct,soIgnoreCertCNInv
    try
       writ('requrl: '+wideFormat(AURL,[lngorig,lngtarget,ate
       httpq.Get(wideFormat(AURL,[lngorig,lngtarget,atext]));
       writ('http get status '+itoa(httpq.response.statuscode
       urlback:= httpq.response.ContentAsUTF8String;
       writ('ctype transback: '+(httpq.response.Contenttype))
       jo:= TJSON.Create();
       jo.parse(urlback);
```

```
        jarr:= jo.JsonArray;
        if httpq.response.statuscode2=200 Then
           result:=jarr[0].decode(urlback)
        else result:='REST API Failed:'+
                  itoa(Httpq.response.statuscode2)+httpq.respo
     except
        writ('EWI_HTTP: '+ExceptiontoString(exceptiontype,exce
     finally
        httpq.free;
        httpq:= Nil;
        jo.free;
     end;
   end;
```

We use a Google API endpoint.

https://clients5.google.com/translate_a/t?client=dict-chrome-ex&sl=%s&tl=%s&q=%s;

You can combine maXbox with the (official or unofficial) Google Translate API for many more things than just "translate one text and show the result".scribd+4. The last step is to gain information concerning the detected country, city or location with the Country API which

untry in the world. For flag images of countries, check out the Country Flag API instead.

Country API – API Ninjas

```
const URL_APILAY_GEOCOUNTRY= 'https://api.api-ninjas.com/v1/c

function APIGetGeoCountry(AURL,url_name,aAPIKey: string): str
var encodedURL:String;
 begin
   encodedURL:= Format(AURL,[urlencode(url_name)]);
   with THttpRequestC.create(self) do
     try
       writ('reqsend: '+encodedurl)
       headers.add('X-Api-Key:'+aAPIkey);
       if Get(encodedURL) then
          result:= (Response.ContentAsString)
       else Writeln('APIError '+inttostr(Response.StatusCode2
     except
       writeln('HTTPS: '+ExceptiontoString(exceptiontype,exce
     finally
       free;
     end;
 end;
```

The Geo Code

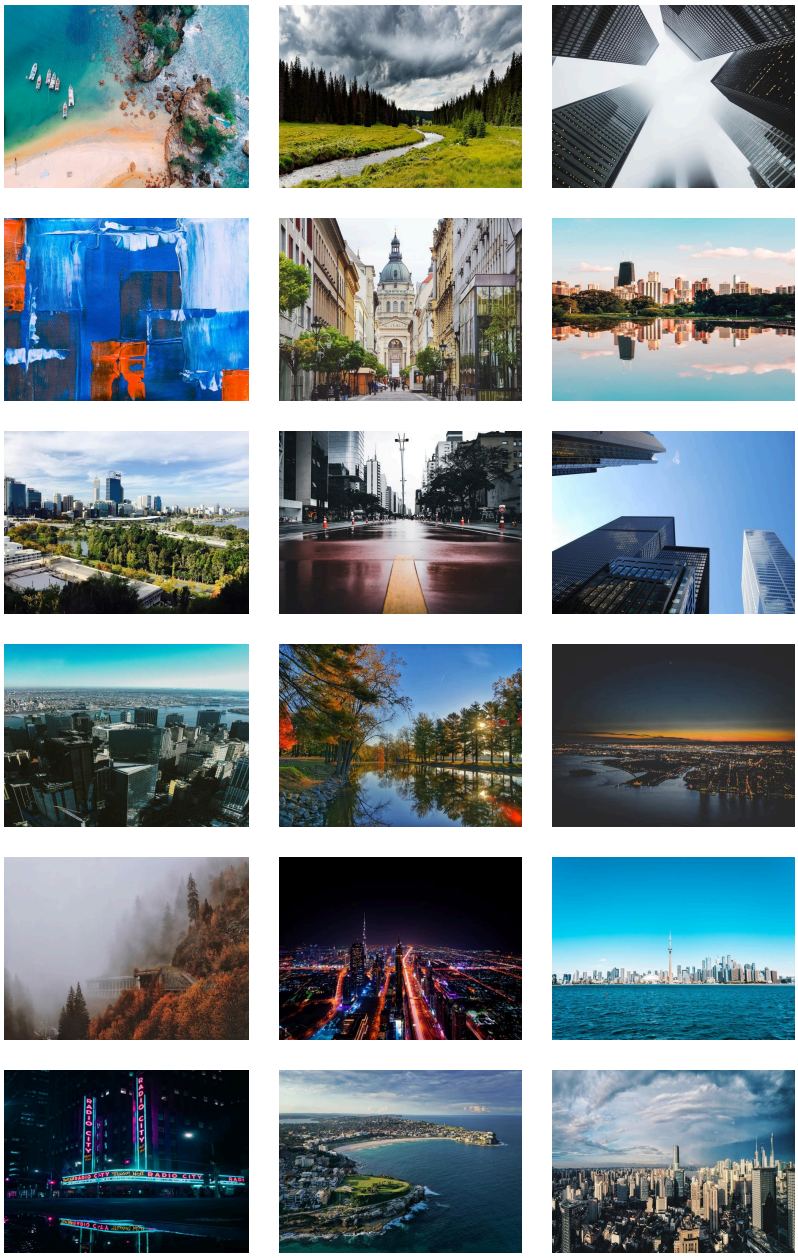
Most of the code is Delphi, Python4Delphi or Python itself (3.14.3 (tags/v3.14.3:323c59a, Feb 3 2026, 16:04:56) [MSC v.1944 64 bit (AMD64)]). As

a last note is the Image Random Generator which generates random images for all your placeholder and design needs. It supports custom sizes as well as custom image categories as in my examples GEO Data.

You also need an API Key which has the format

```
headers.add('X-Api-Key:'+aAPIkey);
```

otherwise you get the error: debug: 179-
The requested header could not be located. 939 err:20
HTTPS: Exception: The requested header could not be located.



Random Image API – API Ninjas

Conclusion

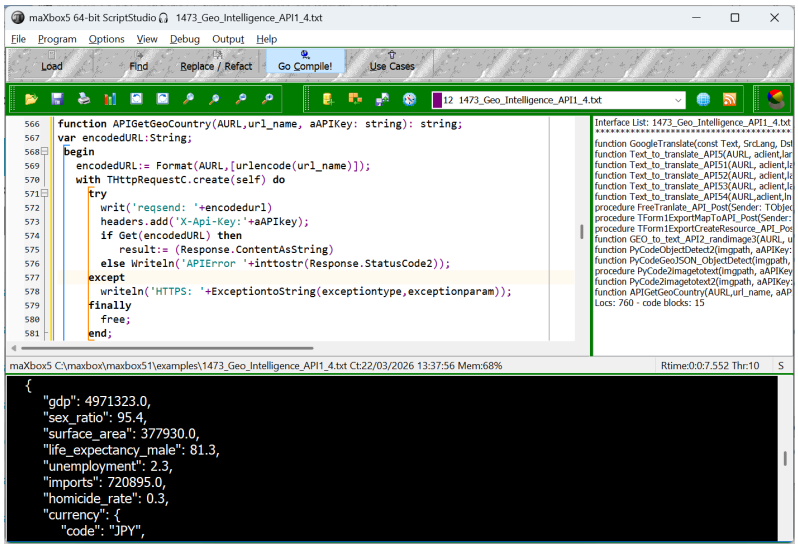
The use of artificial intelligence in the GIS context as the name GEOINT suggests has evolved in recent years from specialized applications to a wide range of innovative and useful technologies. For years ago, GeoAI or GEO Intelligence was mainly used in the field of computer vision for object and pattern recognition in aerial or railway images and for classifying data point clouds. Since then, the range of applications has expanded significantly,

including the field of evaluating environmantal time series to identify trends, meteo or anomalies.

References:

GeoJSON in OSM – Trans Europ Express

Free Google Translator API II – maXbox6



mX5.2.9 API Process Pipeline

Posted in Code, Data Science, Geography, Languages, Machinelearning, maXbox, Python

Leave a comment

← Rosetta Code